

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Robô com Comportamento Definido por Fios Virtuais e Interface em *Tablet*

Vasco Manuel Nunes Pinto

MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Orientador: Armando Jorge Sousa

9 de Agosto de 2018

Resumo

Nesta dissertação procura-se criar um robô para fins educativos variados, na sequência do robô REDi da UP e do RODA da UA mas a custo interessante e fácil de replicar. Deve utilizar primordialmente componentes disponíveis para o consumidor genérico, ser fácil de manter, sem ter elevadas dependências de infraestruturas externas. A utilização incluirá tutoriais e demonstrações em escolas, feiras, visitas e Universidade Júnior, situações onde não é interessante ter PCs mas onde diversos tipos de dispositivos *handheld* fazem sentido.

Foram estudadas diversas soluções de mercado e as alterações para cumprir os requisitos listados no texto da dissertação.

Foi inicialmente abordada a solução "REDi2" que seria um robô um projetado de raiz com componentes comuns e partes mecânicas impressas a 3D.

Mais tarde optou-se pela solução "REDi3" que adapta o robô comercial Whaveshare Alphabot 2, que, na versão utilizada tem por base o conhecido computador de baixo custo Raspberry Pi, concretamente na versão 3B. Este computador permite executar todo o código de controlo do robô e graças ao S.O. Linux genérico e permite que a visualização seja feita em qualquer dispositivo *handheld* (através do protocolo standard VNC). O software embarcado no robô tira proveito parcial do software (anterior a este trabalho) RediSim e utiliza uma mistura de Lazarus / FreePascal e Python. Quando o Robô é ligado, ele inscreve-se numa página web que pode ser livremente consultada com o objetivo de dispensar o monitor local. Esta abordagem permite utilizar redes anteriormente configuradas incluindo eduroam.

As limitações do Waveshare Alphabot 2 e do RPi serão contornadas com recurso a sensores virtuais por intermédio da utilização da câmara que o robô inclui, suportes desenhados neste trabalho e um par de espelhos acrílicos que constituem um sistema de visão catadióptrico. Este sistema de visão permite simular diversos "sonares" sem hardware, sem partes móveis nem custos associados porém exige alguma caracterização do "fundo" (*background*) do ambiente do robô e contraste dos objetos. O projeto foi concebido para um robô flexível e sem necessidade de elevada precisão mas ainda assim procurando minimizar as partes cegas tapadas pelo próprio robô e com um alcance de cerca de meio metro para a frente e meio metro para trás do robô.

Abstract

The purpose of this dissertation is the creation of a robot for multiple educational purposes following the REDi robot from UP and RODA from UA models, but with a more interesting cost and easier replication processes. The goal is to build a robot not only using components available for the general consumer but also the robot should be easily maintained and mostly independent of external infrastructures. Its utilization is directed for situations in which the utilization of handheld devices is more convenient than the presence or use computers (PCs), such as tutorials and demonstrations in schools, fairs, exhibitions, visits and in Junior University ("Universidade Júnior").

Multiple market solutions had been studied and are listed in the text of this dissertation, as well as alterations required to accomplish the listed requirements.

Initially, the "REDi2" solution is addressed and is stated that the resulting robot would be projected from scratch with common components and mechanical parts printed in 3D.

Later, "REDi3" solution is chosen and this robot is an adaptation of the commercial robot WaveShare AlphasBot 2, which is based on the low-cost computer Raspberry Pi. This computer allows the total execution of the robot control code. Due to its O.S. the visualization is possible on any handheld device through the well known VNC protocol. The embedded software on the robot takes partial advantage of RediSim, earlier to this work, and uses a mixture of Lazarus/FreePascal and Python.

When turned on, the robot self enrolls in a web page, publically visitable. This allows users to connect in known networks, even eduroam and save the problems and cost of adding a physical display.

AlphasBot 2 and RPi limitations are surpassed with the utilization of virtual sensors based on the camera included on the robot. Supports designed in this work and a pair of acrylic mirrors form a catadioptric vision system and this vision system allows the simulation of multiple, hardware-free "sonars" without mobile parts and its associated costs. The vision system assumes high luminance "white" background and contrasting objects.

The project was designed for a flexible robot that doesn't require high precision but with the ability to account for the minimization of the blind spots created by the robot structure itself. Also, the range of the robot is planned to be approximately half a meter to its front and to its back.

Agradecimentos

Em primeiro lugar quero agradecer ao Professor Armando Sousa, meu orientador, pela ajuda ao longo deste período, pelo tempo investido e pelo conhecimento transmitido que possibilitou que esta dissertação fosse possível.

Em primeiro lugar quero agradecer aos meus pais por terem feito tudo o que podiam para que fosse possível chegar ao ponto de escrever estes agradecimentos, pois sem eles não estaria a escrever este texto no lugar onde estou. Em particular e em primeiro lugar, quero agradecer à minha mãe, por nos últimos cinco anos me lembrar todos os dias que existe um curso ao qual é preciso trabalhar, pois existiram momentos em que foi difícil manter esse foco e por me motivar a manter-me nele, porque o amor de mãe é o maior amor do mundo. Em particular e em primeiro lugar, quero agradecer ao meu pai por apostar na abordagem contrária mas também eficaz. Sobre uma mesa de bilhar, em jogos bem disputados e conversas de café, a preocupação com o meu futuro nunca fugia da mesa, nem quando o ocasional chouriço acontecia. Agradeço-lhes e espero que possam olhar dentro de pouco tempo para uma missão cumprida de criar um engenheiro à moda do Marco.

Quero também em primeiro lugar, deixar uma palavra ao Bu, que desde que me ia buscar à escola em pequenino que me ensinou a ser engenheiro, apesar de ser contabilista e apenas partilharmos os números. Mas foi nas letras que me ensinou a ser engenheiro, porque nos trabalhos de casa de português, era mais eficiente e rápido fazer um ditado e depois fazer-me corrigir os erros, do que fazer cópias dos textos do livro. Ao seu lado e também em primeiro lugar quero agradecer à Bó, que apesar de não fazer ideia do que ando a fazer na faculdade, se preocupou comigo, desde a minha alimentação até à velocidade com que fazia as viagens entre o Marco e Porto.

Agora, em primeiro lugar, quero agradecer ao meu irmão por me mostrar que basta força de vontade para fazer o que quer que seja e por me mostrar que basta querer muito e trabalhar para a conseguir, e que apesar de ser mais novo, foi um exemplo para mim.

Em primeiro lugar, quero agradecer aos meus sete. Não consigo neste espaço referir todas as alturas marcantes em que me ajudaram, pois necessitava de um tamanho de uma dissertação para falar delas todas, mas tenho que salientar sem referir ou pormenorizar, o que cada um de vocês me deu para me fazer o que sou hoje. Foram cerca de cinco anos de prática a traçar um caminho que nos fez crescer uns com os outros, quer nas aulas, quer e depois delas. Em especial, sem vocês não conseguia passar algumas ondas neste percurso e por isso, missão cumprida.

Por fim, em primeiro lugar, quero deixar uma palavra aos que foram meu exemplo na faculdade, por me orientarem e tratarem de me mostrar o caminho correto neste curso e fazerem de mim mais engenheiro. Por último e por isso em primeiro lugar, quero deixar uma palavra neste texto para juntar a outras que fui deixando durante este percurso, a quem eu tentei ser tanto quanto ou mais o que outros foram para mim.

*"Vai com cuidado,
cento e oitenta, duzentos."*

José Soares Nunes

Conteúdo

1	Introdução e Contexto	1
1.1	Contexto	1
1.2	Motivação	1
1.3	Objetivos	1
1.4	Estrutura do Documento	2
2	Estado da Arte	3
2.1	Competições	3
2.1.1	<i>First Lego League</i>	3
2.1.2	<i>RoboParty</i>	3
2.1.3	<i>Festival Nacional de Robótica</i>	4
2.1.4	<i>RoboCup</i>	4
2.2	Demonstrações	5
2.2.1	ICARSC	5
2.2.2	Universidade Júnior	5
2.3	Robótica e Educação	6
2.3.1	USARSim	6
2.3.2	e-puck	6
2.4	Mercado	8
2.4.1	REDi	10
2.5	Ferramentas e Tecnologias	11
2.5.1	Lazarus	11
2.5.2	Autodesk Fusion 360	12
2.5.3	OpenCV	12
3	Caracterização do Problema e propostas de solução	13
3.1	Requisitos do Sistema	13
3.1.1	Desafios	13
3.2	Soluções Propostas	15
3.2.1	Desenvolvimento de um robô de raiz: <i>REDi2</i>	15
3.2.2	Melhoramento de um robô: REDi3	17
3.2.3	Sonares Virtuais	32
3.2.4	Utilização do Simulador <i>RediSim</i>	33
3.2.5	Modulo de Controlo dos Motores	33
3.2.6	Interface, Registo e Consulta de Robôs	34
3.2.7	Utilização da Câmara Original do AlphaBot2	35
3.2.8	Bateria	36

4	Conclusões	39
A	Redi2 - Custos dos Componentes	41
B	Redi3 - Custos dos Componentes	43
	Referências	45

Lista de Figuras

2.1	Cam Jam EduKit 3	9
2.2	Alphabot 2	9
2.3	REDi	9
2.4	RODA	9
2.5	LEgo Mindstorms EV3	9
2.6	MAkeblock mbot v1.1	9
2.7	Base superior do REDi	11
2.8	RediSim: Cavilhas	12
2.9	RediSim: Simulação do movimento do robô	12
3.1	Vistas do modelo CAD	15
3.2	Arquitetura do REDi2	16
3.3	AlphaBot2-Base: Vista da parte superior	17
3.4	AlphaBot2-Base: Vista da parte inferior	17
3.5	AlphaBot2-Pi: Vista da parte superior	18
3.6	AlphaBot2-Pi: Vista da parte inferior	18
3.7	AlphaBot2	18
3.8	Circunferências usadas para referência	19
3.9	Vista da câmara original	20
3.10	Suporte do espelho convexo	21
3.11	Suporte da câmara horizontal	21
3.12	Suporte do espelho convexo	21
3.13	Suporte da câmara horizontal	21
3.14	AlphaBot2 com o espelho convexo	22
3.15	Imagem captada utilizando o espelho convexo	22
3.16	Plano XY sobre o qual foi feito o corte	23
3.17	Modelo geométrico da solução com espelhos em 'V', no plano XY	23
3.18	Representação geométrica 1	24
3.19	Representação geométrica 2	24
3.20	Representação geométrica 3	25
3.21	Representação geométrica 4	25
3.22	Suporte do espelho em 'V'	27
3.23	Suporte da câmara horizontal para o espelho em 'V'	27
3.24	Suporte do espelho em 'V'	28
3.25	Suporte da câmara horizontal para o espelho em 'V'	28
3.26	AlphaBot2 com os espelhos em 'V'	28
3.27	Imagem captada usando os espelhos em 'V'	28
3.28	Representação do Espaço Visível	28

3.29	Diagrama de controlo	29
3.30	Exemplo de imagem capturada	30
3.31	Divisão da imagem por zonas	30
3.32	Área 1 da imagem original	30
3.33	Tons de cinzento	30
3.34	Imagem binária	30
3.35	Contornos da Área 1	30
3.36	Contornos da retaguarda	31
3.37	Contornos da direita	31
3.38	Contornos do meio	31
3.39	Contornos da esquerda	31
3.40	Linhas de tendência	32
3.41	Gráfico percentual da distância	33
3.42	Interface com recurso a VNC	34
3.43	Interface com recurso a SSH	34
3.44	Imagem captada com a configuração para a câmara original	36
3.45	Suporte das baterias	37
3.46	Suporte colocado	37
3.47	Suporte em pose de substituição	37

Lista de Tabelas

2.1	Comparação entre robôs Existentes	8
A.1	Preço dos componentes da Solução Proposta	41

Abreviaturas e Símbolos

DEEC	Departamento de Engenharia Eletrotécnica e de Computadores
FEUP	Faculdade de Engenharia da Universidade do Porto
ICARSC	International Conference on Autonomous Robot Systems and Competitions
USARSim	Unified System for Automation and Robot Simulation
EPFL	Escola Politécnica Federal de Lausana
UDP	User Datagram Protocol
IP	Internet Protocol
SSH	Secure Shell
VNC	Virtual Network Computing
PHP	PHP:Hypertext Preprocessor
RAD	Rapid Application Development
USB	Universal Serial Bus
CAD	COmputer-Aided Design
3D	3 Dimensions
PWM	Pulse Width Modulation

Capítulo 1

Introdução e Contexto

Com este documento pretende-se demonstrar o processo de elaboração de um robô, com uma interface num *tablet*, capaz de ser utilizado como ferramenta de ensino em escolas e de demonstrações de robótica.

1.1 Contexto

Esta dissertação está inserida no contexto da robótica e a sua utilização como ferramenta de ensino. Atualmente, existe um vasto leque de ofertas e soluções para várias aplicações dentro da área educativa. O robô desenvolvido enquadra-se na aplicação destas tecnologias como método de formação para estudantes do ensino básico, no ramo curricular da programação.

1.2 Motivação

Com o aumento do uso da robótica no quotidiano e, pelo facto de cada vez mais cedo existir interação com aplicações autónomas, cria-se a necessidade de introduzir conceitos básicos de robótica e de ciência de computadores a gerações cada vez mais jovens.

É notório o crescimento e investimento universal nas áreas de engenharia, particularmente no ramo da robótica. As novas técnicas desenvolvidas diariamente a um ritmo acelerado, associado às capacidades que estas possuem e as suas aplicações, levam a que um número considerável de jovens desenvolva um interesse antecipado por este mundo.

Então, com o intuito de encontrar uma resposta para esta demanda sentiu-se a necessidade de criar um robô com aplicações educativas, sendo esta a principal motivação para a génese desta mesma dissertação.

1.3 Objetivos

A utilização de robôs para fins educativos deve ser simples e intuitiva, com o objetivo de qualquer pessoa sem bases em programação o possa entender. Assim sendo, um dos principais

objetivos é permitir o desenvolvimento de capacidades basilares que permitam abrir portas a este meio. Outro claro objetivo desta dissertação é desenvolver um robô simples, fácil de produzir e economicamente viável para que possa ser produzido em grandes quantidades e chegar a escolas de todo o país.

Pretende-se, ainda, que este robô seja utilizado em demonstrações do Departamento de Engenharia Eletrotécnica e de Computadores (DEEC) e da Faculdade de Engenharia da Universidade do Porto (FEUP) através de parcerias com outras entidades.

1.4 Estrutura do Documento

Para além da Introdução e Contexto, a presente dissertação engloba mais quatro capítulos. No capítulo 2, é descrito o Estado da Arte e são apresentados trabalhos relacionados. No capítulo 2.5, são abordadas ferramentas e tecnologias utilizadas neste projeto. No capítulo 3, é abordado o problema e as soluções propostas para a resolução do mesmo. Finalmente no capítulo 4 são descritas as conclusões e resultados obtidos do trabalho desenvolvido nesta dissertação.

Capítulo 2

Estado da Arte

Neste capítulo serão abordados projetos existentes nesta área da Engenharia relacionados com a robótica.

2.1 Competições

2.1.1 *First Lego League*

Anualmente, é lançado pela *First Lego League* um desafio baseado num tema real aplicado ao mundo da ciência como, por exemplo, nanotecnologia, clima ou qualidade de vida de pessoas com incapacidades, que já foram tópicos em competições passadas e, tem como objetivo, ser superado pelas equipas participantes usando material da *Lego*.

Cada desafio consiste em três partes, *Robot Game*, *Project* e *Core Values*. Estas exploram e qualificam de acordo com as regras estabelecidas e os requisitos impostos para cada parte da competição. Cada equipa, com o máximo de dez crianças e no mínimo orientada por dois adultos, participa programando um robô autónomo capaz de superar cada uma das partes [1].

2.1.2 *RoboParty*

A *RoboParty* é um evento de três dias consecutivos, de vinte e quatro sob vinte e quatro horas, com um intuito pedagógico, que tem como objetivo ensinar a criar robôs móveis autónomos, de forma didática e apelativa. Durante o evento, pessoas qualificadas acompanham o desenvolvimento dos projetos e ensinam conceitos essenciais para a construção de um robô. O evento começa com uma formação de modo a que os participantes tenham uma introdução à programação de robôs, eletrónica e à construção mecânica. No final da formação é disponibilizado um kit básico, desenvolvido pela "*botnroll*" e pela Universidade do Minho (UM), com opções para expandir o robô para cada desafio.

O evento é composto por vários desafios para as equipas participantes, que com a devida expansão do kit, constroem um robô para a superação do desafio [2].

A *Race of Champions* é uma corrida de robôs autônomos numa pista em que o robô deve seguir um percurso delineado por uma linha preta no chão fazendo uso da expansão do kit, seguidor de linha. O robô deve seguir e conseguir dar a volta o mais rapidamente possível, com penalizações de tempo caso este infrinja alguma regra ou desqualificação se o robô sair completamente de cima da linha [3].

No desafio de obstáculos, o robô deve utilizar o kit disponível para a prova fazendo a programação do robô, percorrer um labirinto e sair deste no menor tempo possível, evitando colisões com as paredes [4].

O desafio de dança promove a criatividade e junta várias equipas para decorarem os seus robôs e criarem uma coreografia de vários robôs que, ao ritmo de uma música à sua escolha, deva ser executada perante um júri que avalia a criatividade e a prestação dos robôs [5].

2.1.3 *Festival Nacional de Robótica*

O *Festival Nacional de Robótica* é, atualmente, o maior encontro científico com competições de robótica em Portugal. Anualmente, desde o ano de 2001, a Sociedade Portuguesa de Robótica em parceria com instituições escolhidas para a organização do festival, promovem o evento em diferentes cidades do país proporcionando uma distribuição geograficamente equilibrada. Tem como objetivo a divulgação da Ciência e Tecnologia desenvolvida em Portugal, motivando os alunos de escolas básicas e secundárias para a área da robótica, dando um contributo para o desenvolvimento da investigação da robótica e automação em Portugal, e efetuando o apuramento das equipas portuguesas para a competição mundial de robótica, a *RoboCup*.

Este festival é composto por três frentes principais, competições robóticas, encontro científico e demonstrações. As competições são constituídas pela Liga Júnior e Liga Sénior e abordam as áreas do futebol robótico, condução autónoma, construção livre, entre outras [6].

2.1.4 *RoboCup*

Realizada pela primeira vez em 1997 no Japão e, desde então, anualmente realizada em diferentes países, a *RoboCup* conta nos últimos anos com cerca de três mil participantes de cerca de quarenta países.

"Em meados de século 21, uma equipe totalmente autónoma de robôs humanóides jogadores de futebol deve vencer um jogo contra o time de humanos campeão da última Copa do Mundo da FIFA, utilizando as regras da FIFA.", conta o mais recente vencedor do Campeonato do Mundo.

Esta competição tem como objetivo incentivar e promover o desenvolvimento da inteligência artificial através de um objetivo a longo prazo, que terá impacto em cada sub objetivo que é alcançado. Existem, então, várias ligas dentro desta competição. A *RoboCupSoccer* que é um jogo de futebol adaptado a robôs humanoides; a *RoboCupRescue* que tem como objetivo desenvolver capacidades de resposta de robôs a cenários de emergência; a *RoboCupHome* que tem como intuito a criação de robôs capazes de assistir em aplicações domésticas; a *RoboCupIndustrial* com o objetivo de desenvolver tecnologias capazes de ser aplicadas ao mundo industrial, do trabalho

e logística; e a *RoboCupJunior* que é orientada para a educação e com o objetivo de introduzir estudantes que ainda não possuem as capacidades para participar nas restantes ligas [7].

2.1.4.1 5DPO: Futebol Robótico

5DPO é nome do projecto de futebol robótico existente na FEUP, fundado em 1997 por um grupo de docentes que tinha como objetivo garantir a sua participação no encontro *RoboCup*.

O futebol, devido às suas características únicas permite facilmente testar a destreza e habilidade dos robôs e, visto ser um desporto de equipa permite também por a prova a coordenação entre robôs, bem como a sua capacidade de contornar barreiras.

Para tal, são implementados algoritmos complexos que dizem ao robô como agir nas diferentes situações. Estes podem ser aplicados de forma semelhante noutras áreas, tais como, operações de busca e salvamento, onde os diferentes elementos têm de trabalhar em conjunto para um mesmo objetivo.

Devido ao facto de este desporto robótico desenvolver inúmeras tecnologias como cooperação, comunicação, raciocínio em tempo real e aprendizagem, estas atividades são um forte incentivo para o desenvolvimento científico-tecnológico que pode ser posteriormente para diversas áreas como a indústria, a medicina, entre outras.

2.2 Demonstrações

2.2.1 ICARSC

A *International Conference on Autonomus Robot Systems and Competitions* (ICARSC) é um fórum internacional que reúne investigadores de várias áreas da robótica para discutir sobre esta área de investigação. De forma a ser possível comparar avanços no Estado da Arte, são realizadas competições para avaliar as capacidades dos robôs realizarem certas tarefas.

Esta conferência é promovida pelo *Institute of Electrical and Electronics Engineers* (IEEE) em conjunto com outras organizações que patrocinem o evento localmente [8].

2.2.2 Universidade Júnior

A Universidade Júnior é um programa da Universidade do Porto que permite aos estudantes do ensino básico e secundário conhecerem projetos de várias áreas da Universidade do Porto. Desta forma, é-lhes permitido estarem em contacto com técnicas e abordagens no terreno participando em atividades lúdicas durante o verão e dando a conhecer a realidade do Ensino Superior.

Um dos projetos que a Universidade Júnior oferece é "À descoberta da Robótica" promovido pelo Departamento de Engenharia Eletrotécnica e de Computadores (DEEC), onde os estudantes, do décimo ao décimo segundo ano de escolaridade, entram em contacto com a robótica. Através de demonstrações da mesma e desafios que passam por simuladores, até à programação de um robô para a execução de tarefas propostas que desafiam os estudantes a desenvolverem capacidades na

área da robótica, é aberta a porta à possibilidade de poderem participar em provas nacionais e internacionais, que são exemplos, o *Festival Nacional de Robótica*, e a *RoboCup* [9].

2.3 Robótica e Educação

2.3.1 USARSim

O *Unified System for Automation and Robot Simulation* (USARSim) é um simulador baseado num motor de jogo que permite aos utilizadores construir e testarem os seus projetos de robótica. Dada a sua versatilidade e facilidade de acrescentar componentes, modelos ou robôs novos e a fácil configuração do simulador, permite que este ambiente seja uma boa maneira de desenvolver novos projetos a partir do zero ou baseado em projetos já existentes, que para propósitos educativos e de introdução à robótica, revela ser uma boa alternativa aos robôs do mundo real, pois estes necessitam de equipamento e recursos que por vezes não existem. O simulador fornece também uma avaliação que permite comparar os resultados obtidos no simulador com os resultados obtidos com o sistema no mundo real. Este tem sido amplamente usado como infraestrutura para competições dentro da *RoboCup*, o que cria um grande leque de utilizadores habilitados que eventualmente partilham o seu código desenvolvido com a comunidade científica para uma vasta disseminação [10].

2.3.2 e-puck

O *e-puck* é um robô desenvolvido pela Escola Politécnica Federal de Lausana (EPFL) com o intuito de ser usado no ensino universitário e investigação a um preço acessível ao orçamento de estabelecimentos de ensino. O robô foi desenvolvido para substituir o seu modelo anterior que estava a ser usado nos cursos da EPFL em aulas de robótica, no entanto o *e-puck* permitiu também que passasse a ser usado em aulas de anos prévios, que o seu antecessor não estava a ser utilizado. As funcionalidades do *e-puck* abriram também as portas a que o robô fosse utilizado em aulas adjacentes à robótica, como por exemplo, processamento de sinal, automação, e sistemas embarcados.

Para que este robô pudesse substituir os seus antecessores, teriam que ser cumpridos os seguintes requisitos, que até à altura não existia no mercado um robô capaz de responder a todos os listados:

- O seu tamanho deve ser reduzido de modo a permitir a sua utilização numa mesa de uma sala de aula;
- Oferecer uma vasta gama de possibilidades do ponto de vista de Engenharia e Educacional que permitisse explorar esta ferramenta em bastantes áreas;
- Deve ser intuitivo, com uma interface simples, para que os estudantes consigam explorar com objetividade;

- Preço reduzido, para que possa ser adquirido um grande número de robôs, para satisfazer a necessidade dos estabelecimentos de ensino na relação preço/quantidade;
- Capaz de ser partilhado por vários utilizadores, pois será usado por diferentes estudantes, professores e laboratórios.

O *e-puck* é dotado de oito sensores infravermelhos de proximidade colocados à volta do robô, um acelerómetro, três microfones, uma câmara, dois motores que controlam o movimento de duas rodas de mil *steps* de resolução, uma coluna, um recetor infravermelho que permite o controlo por um comando de televisão, *bluetooth*. O robô tem também a capacidade de lhe ser acrescentado extensões para a utilização em cenários específicos, como uma torre com três câmaras que permite ver uma visão ampla e, para exercícios em que os sensores de proximidade não são suficientes, pode ser adicionado um *scanner* rotativo de infravermelhos [11].

2.4 Mercado

A tabela 2.1 é uma tabela comparativa com alguns dos robôs existentes no mercado e desenvolvidos em ambiente acadêmico.

Tabela 2.1: Comparação entre robôs Existentes

	Locomoção	Componentes	Preço
<i>Cam Jam EduKit 3</i> [12]	2 Rodas	2 Motores / Controlador do motor / 2 Rodas / 1 Roda livre / 1 <i>Breadboard</i> / Caixa para pilhas / Sensor de distância ultrassom / Seguidor de linha / Resistências e cabos	20,40 EUR
<i>AlphaBot 2</i> [13]	2 Rodas	Sensores infravermelhos / Seguidor de linha/ RPI câmara (B) / Cartão micro SD 16 GB / Adaptador 5V/2.5V USB / Comando infravermelhos / Cabo USB A para micro B / Leitor de cartão micro SD	100,30 EUR
<i>REDi</i> [14]	2 Rodas	Sensor de chão / Sensor de linha / Sonares / <i>Touchscreen</i> / Painel de ligações com cavilhas / Bateria / 2 Motores e <i>drive</i> / Arduino mega	
<i>RODA</i> [15]	2 Rodas	Sensores de chão / Sensores de obstáculos / Motores / <i>Encoders</i> / Servo motor / Placa de ligações virtuais	
<i>Lego Mindstorm EV3</i> [16]	2 Lagartas / Rodas / Pernas / Articulado	Sensor de cores / Giroscópio / Infravermelhos / Sonar / Sensor de toque	258,39 EUR
<i>Makeblock mbot v1.1</i> [17]	2 Rodas	Modulo <i>bluetooth</i> / Seguidor de linha/ Roda livre / 2 Motores / Sensor ultrassom / Comando infravermelhos	76,22 EUR



Figura 2.1: Cam Jam EduKit 3

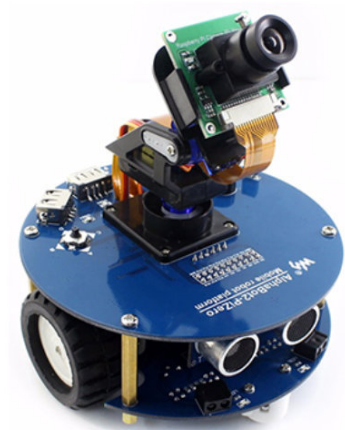


Figura 2.2: Alphabot 2

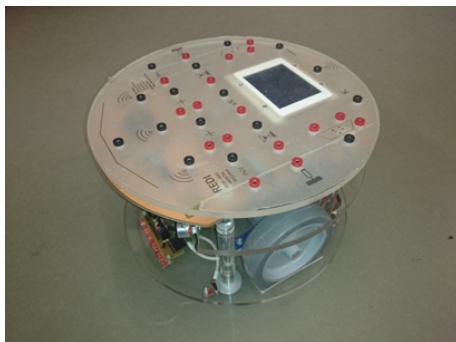


Figura 2.3: REDi

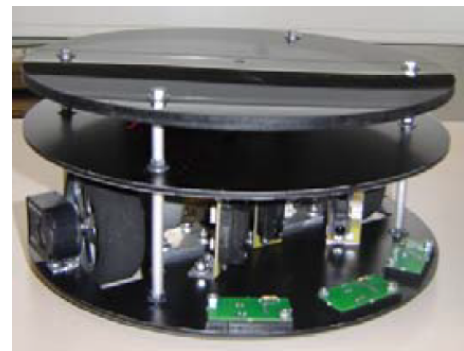


Figura 2.4: RODA



Figura 2.5: LEgo Mindstorms EV3

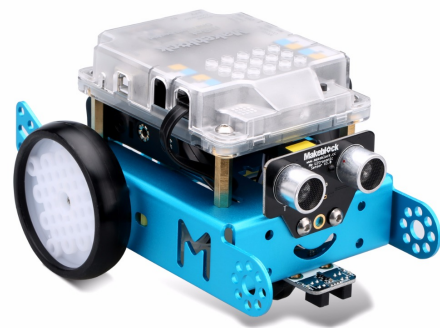


Figura 2.6: MAkeblock mbot v1.1

2.4.1 REDi

O projeto do robô REDi[14] é criado no âmbito da sua utilização na Universidade Júnior, particularmente no curso de Robótica do Departamento de Engenharia Eletrotécnica e de Computadores. O projeto é criado com o intuito de construir um robô adaptado do RODA[15], que tem como propósito desenvolver as capacidades dos mais jovens no ramo da programação e da robótica.

O robô possui um conjunto de sensores e atuadores que podem ser conectados com a utilização de cavilhas, que permitem a ligação através de blocos lógicos que utilizam um Arduino Mega para calcular qual deve ser o valor da saída. Este tipo de programação transparece para o utilizador um ligação direta entre sensores e atuadores que torna a lógica mais simples e intuitiva.

O REDi possui quatro sensores infravermelhos QTR-1A, que tem como objetivo deteção do chão. Estes sensores podem ser usados numa configuração que permita que o robô se mantenha dentro de uma circunferência desenhada no chão.

Tem também um sensor de linha que permite a deteção de uma linha preta num fundo claro e a posição relativa ao centro do robô. É utilizado o QTR-8A, constituído por oito pares emissor recetor infravermelho que funciona de 0V a 5V que é nulo a reflexão máxima numa superfície branca.

Possui quatro sonares do tipo SRF-08 que servem para a deteção de obstáculos à frente, à esquerda, à direita e atrás do robô e permitem que o REDi se desvie de obstáculos, siga paredes e resolver um labirinto. Estes sonares foram configurados para terem um alcance de cinquenta centímetros.

No REDi está presente no seu topo uma base com cavilhas vista na figura 2.7, que servem para determinar quais as interligações efetuadas entre blocos e podem corresponder a entradas e saídas destes, permitindo a criação de um programa que controla o comportamento do robô.

Os blocos permitem a simulação das seguintes funções aritméticas:

- Definir um offset através de uma constante de tempo
- Inversor do sinal de entrada
- Somador de dois sinais de entrada
- Multiplexador de dois sinais de entrada
- Timer determinado pelo utilizador
- Multiplicador de uma entrada por uma constante definida pelo utilizador

Tem também na sua base superior um *touch screen* que permite visualizar sinais escolhidos previamente, a paragem da execução do programa atual e efetuar configurações.

O robô é dotado de dois motores EMG30 com uma drive MD25, que recebem um valor calculado internamente que indicam o sentido e velocidade a que estes devem rodar. Estes possuem encoders que permitem controlar de forma independente.

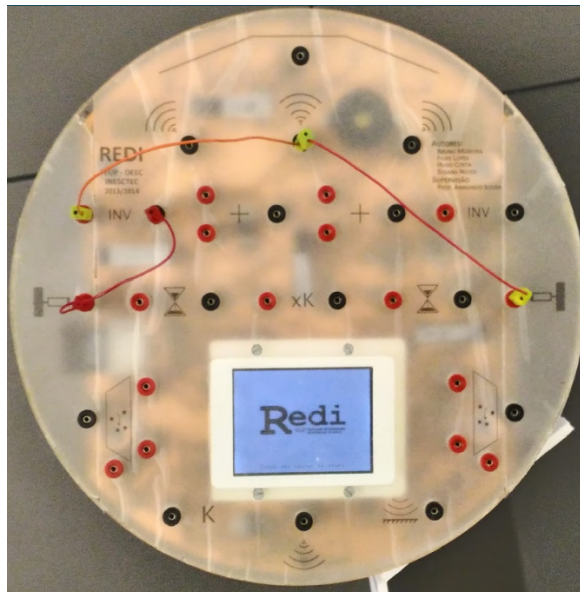


Figura 2.7: Base superior do REDi

2.5 Ferramentas e Tecnologias

2.5.1 Lazarus

Lazarus é um *software open source*, suportado por várias arquiteturas e sistemas operativos, como Windows, Linux e Mac. É um ambiente de desenvolvimento integrado orientado para o compilador *Free Pascal* e também compatível com o *Delphi*, percebendo e compilando a sua sintaxe.

Para além de permitir criar aplicações com mais facilidade e rapidez fazendo uso da técnica RAD (*Rapid Application Development*), permite criar aplicações com uma interface.

2.5.1.1 RediSim

O *RediSim* é um simulador baseado no robô *REDi* desenvolvido em *Lazarus*.

O *REDi* possui na sua base superior um conjunto de cavilhas que permite ao utilizador fazer ligações virtuais entre os componentes. O *RediSim*, também permite a configuração destas ligações recorrendo a uma interface gráfica semelhante à base superior do *REDi*, como mostra a figura 2.8.

Para a configuração das ligações virtuais efetuadas na interface gráfica, o *RediSim* efetua o processamento das entradas, e retorna os valores das velocidades e dos sentidos que devem possuir cada um dos motores.

O resultado desta simulação pode ser observado, recorrendo à interface representada na figura 2.9. Nesta janela é possível criar um mapa com linhas que representam paredes que os sonares do *REDi* detetam e, conseqüentemente, usando os valores retornados de velocidade e sentido dos motores, podemos observar o deslocamento do robô.

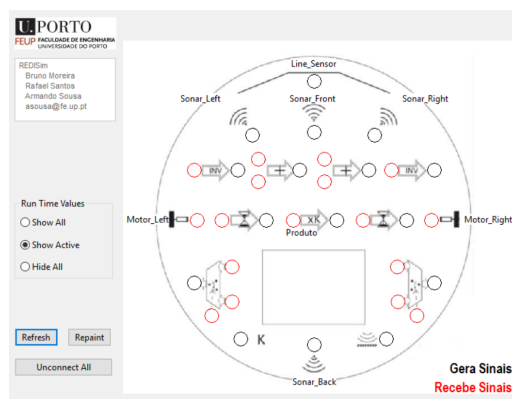


Figura 2.8: RediSim: Cavilhas

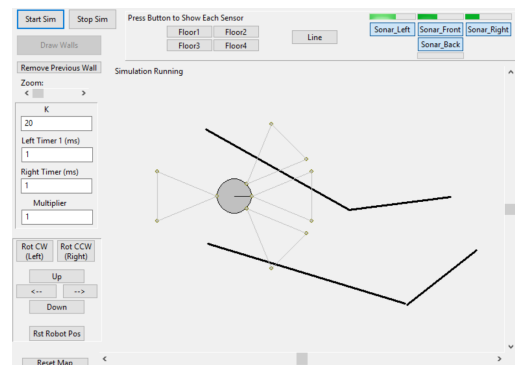


Figura 2.9: RediSim: Simulação do movimento do robô

2.5.2 Autodesk Fusion 360

O *Autodesk Fusion 360* é atualmente uma ferramenta de modelização de objetos a três dimensões com uma comunidade crescente de utilizadores. Este *software* permite a criação de um modelo *CAD* e a sua visualização num ambiente de simulação, o que permite ao utilizador ter uma visão mais próxima da realidade das peças em desenvolvimento. O *Autodesk Fusion 360*, permite a exportação do modelo *CAD* para um formato de ficheiro que permita a sua impressão recorrendo a uma impressora 3D.

2.5.3 OpenCV

O *OpenCV* é uma biblioteca *open source* e *cross platform* de funções, utilizadas para o desenvolvimento de aplicações na área de visão computacional e processamento de imagem em tempo real.

Capítulo 3

Caracterização do Problema e propostas de solução

3.1 Requisitos do Sistema

O sistema deve ser capaz de ser de fácil programação e intuitivo, de modo a que o utilizador consiga desenvolver os seus projetos sem conhecimentos profundos de programação. Para tal, deve possuir uma interface simples, que controle o resto do sistema, sem que para isso o utilizador se preocupe com o funcionamento do robô, podendo-se focar na programação a um nível superior, enquanto que as camadas inferiores funcionam autonomamente.

O robô deve ter características que permitam que este seja facilmente replicável para que possam ser construídos cerca de oito robôs. Consequentemente, o custo do projeto deve ter sido em conta e reduzido ao mínimo para que se torne viável a sua produção.

3.1.1 Desafios

A interação do utilizador com a interface do robô, será orientada a que este o programe de modo a que o robô seja capaz de superar desafios. Estes desafios envolvem níveis diferentes de dificuldade, de modo a que o utilizador seja capaz de desenvolver as suas capacidades de programação e evoluir para desafios mais elaborados.

Robô Íman O robô na presença de um obstáculo à sua frente, deve avançar em direção a este, simulando um íman. Utiliza o sonar na sua frente para detetar os obstáculos que controlam os motores para se mover.

Robô que Foge Na presença de um obstáculo, detetado pelos sonares, o robô inverte o sentido dos motores e move-se na direção oposta ao obstáculo.

Robô que se vira O robô quando detetar um obstáculo na sua frente deve rodar sobre si próprio.

Robô no meio Quando o robô detetar um obstáculo à frente e outro atrás, deve colocar-se no meio dos dois a uma distância igual entre os dois obstáculos.

Robô que mantém a distância a obstáculos Se existir um obstáculo à frente do robô, este deve manter uma distância fixa deste e mover-se para manter essa distância.

Robô que evita obstáculos Quando o robô detetar um obstáculo num dos seus lados deve virar na direção oposta de modo a desviar-se do mesmo.

Robô que segue trajeto definido por muros O robô deve andar em frente, e detetando um obstáculo num dos lados ele deve desviar-se e afastar-se continuando a mover-se em frente. Quando detetar obstáculos de ambos os lados, o robô deve seguir em frente pelo meio dos dois mantendo e guardando uma distância igual a ambos os lados.

Robô seguidor de linha O robô deve fazer uso do sensor linha na sua parte inferior para detetar uma linha no chão e seguir o caminho traçado por esta.

3.2 Soluções Propostas

3.2.1 Desenvolvimento de um robô de raiz: *REDi2*

A solução proposta passa por construir um robô capaz de responder aos requisitos enumerados em cima. O *REDi2* possuirá três bases nas quais os componentes serão montados segundo a arquitetura representada na figura 3.2 e a representação gráfica do modelo do robô na figura 3.1

A base inferior será circular com duas aberturas retangulares de modo a que as rodas passem pelas mesmas, mantendo parte das rodas no interior do robô e permitindo um baixo centro de gravidade. As rodas estarão conectadas aos motores e colocadas no meio da base. O controle dos motores será feita por um *Driver*, também colocado na parte superior da base. No inferior desta base, estará colocado o sensor de linha e duas rodas livres para estabilidade.

Na base do meio, na sua parte superior, serão colocados o *Raspberry Pi* e o *Arduino*, bem como os sonares (três na parte da frente e um na parte traseira). Na parte inferior desta base será colocada a câmara que estará ligada ao *Raspeberry Pi*.

A base superior terá um *display* e *leds* que permitirão ver o estado do robô, e botões para a sua configuração.

3.2.1.1 Modelo CAD do *REDi2*

Na figura 3.1 abaixo pode-se ver o modelo do robô feito no software *open source*, *Autodesk Fusion 360*.

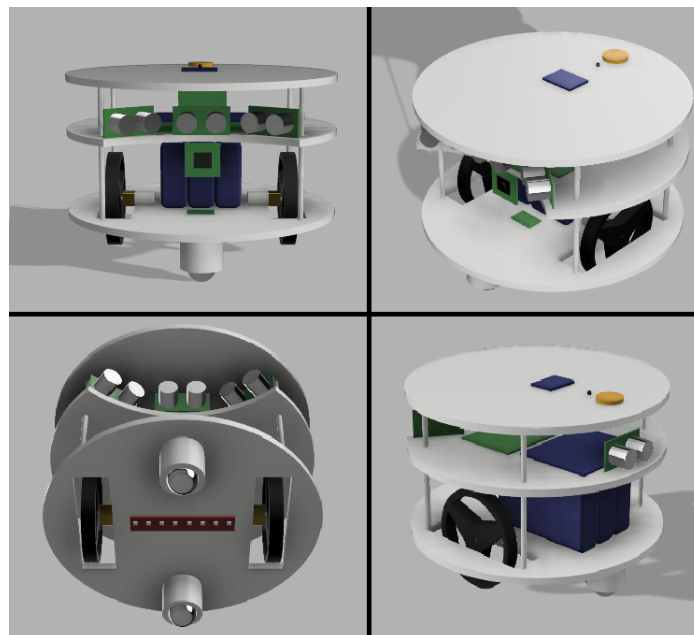


Figura 3.1: Vistas do modelo CAD

3.2.1.2 Arquitetura do REDi2

Na figura 3.2 (abaixo), está representada a arquitetura do robô. O *Raspberry Pi*, alimentado por uma *powerbank*, estará ligado a uma câmara, a um *tablet* através de VNC ou SSH e ligado a um *Arduino* por USB.

O *Arduino*, alimentado por uma *powerbank*, está ligado aos sensores e ao *Motor Driver*, também alimentado por uma *powerbank* e conectado aos motores.

O *Raspberry Pi* será utilizado para controlar o sistema comunicando com o *Arduino*, sendo que o seu propósito é comunicar diretamente com os motores e captar o estado dos sensores para fornecer ao *Raspberry Pi*, onde ocorre o processamento das entradas e saídas.

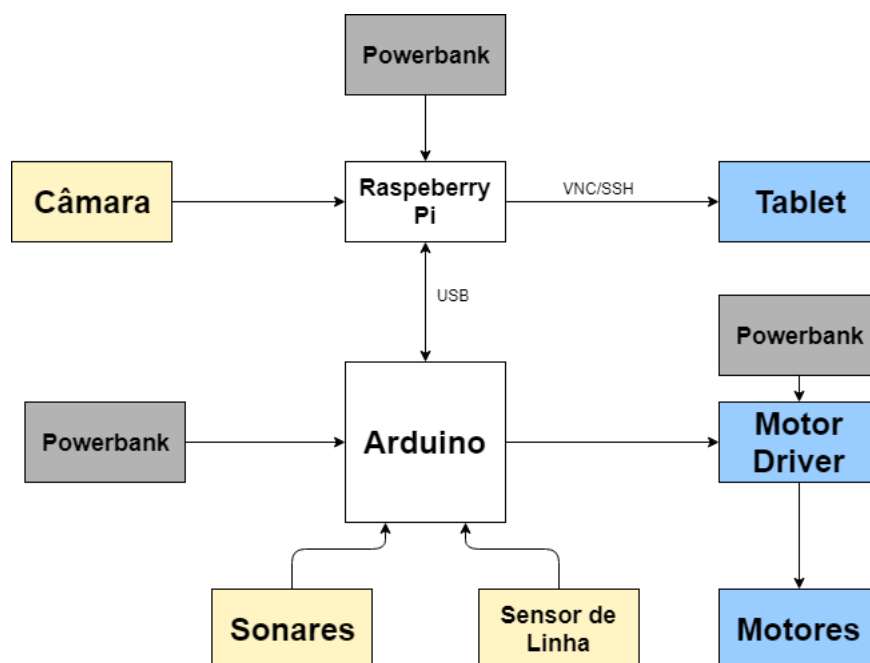


Figura 3.2: Arquitetura do REDi2

3.2.1.3 Custo da solução

No anexo A, pode-se observar uma tabela com os vários componentes necessários para a realização desta solução. O custo total da solução seria de 295,70 EUR, o que não revela ser viável. Apesar de possuir elementos e características que tornariam possível o cumprimento dos requisitos, o custo que esta solução apresenta torna inviável a construção do robô, devido ao facto desta gama de preço ser superior ao pretendido para tornar a solução exequível.

3.2.2 Melhoramento de um robô: REDi3

Em alternativa à construção de um robô de raiz, a utilização de um robô já presente no mercado, que reúna características semelhantes as da solução previamente proposta e que possua um custo inferior é a melhor opção para a solução do problema.

O *AlphaBot2* é um robô que se enquadra na gama de preço desejada. O custo de 100,30 EUR e os elementos presentes neste robô, revelam ser uma opção viável apesar de possuir algumas limitações, que iremos analisar de seguida.

3.2.2.1 Alphabot2

O *Alphabot 2* é um kit composto por um chassis base, o *AlphaBot2-Base*, e por uma das placas adaptadoras que suporta um *Arduino*, um *Raspberry Pi 3 Model B* ou *Raspberry Pi Zero W*, respetivamente a *AlphaBot2-Ar*, *AlphaBot2-Pi* e *AlphaBot2-PiZero*. Possui características bastante comuns noutros robôs que permitem seguir a linha e desviar de obstáculos, e tem integrado *Bluetooth*, *Wi-Fi*, infravermelhos e comando remoto.

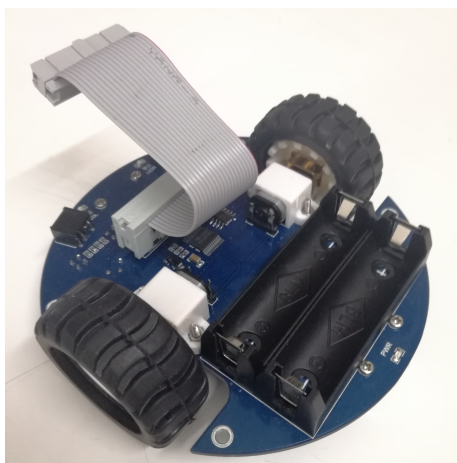


Figura 3.3: AlphaBot2-Base: Vista da parte superior



Figura 3.4: AlphaBot2-Base: Vista da parte inferior

O *AlphaBot2-Base*, na sua parte superior [3.3](#), possui dois sensores infravermelhos ST188, que permitem o desvio de obstáculos, mas que tem no entanto a limitação que apenas podem ser calibrados recorrendo a um potenciômetro, para detetarem um obstáculo a uma dada distância, não fornecendo informação sobre o valor concreto a que o este se encontra, sabendo apenas se se encontra mais próximo da distância calibrada. As duas rodas de 42 mm de diâmetro são movidas recorrendo a dois motores *N20 micro gear motor*, ligados a uma *driver* de dupla ponte em H que permite o controlo em separado dos dois motores. Tem ainda o suporte de pilhas 14500 que alimentam o *AlphaBot2-Base*, a placa adaptadora *AlphaBot2-Pi* e ainda o *Raspberry Pi*, através de um cabo que une o *AlphaBot2-Base* ao *AlphaBot2-Pi*, que fornece a comunicação entre os dois e permite que a Base seja controlada pelo *Raspberry Pi*.

Na parte inferior 3.4, o *AlphaBot2-Base* possui cinco sensores infravermelhos ITR20001/T, que permitem que o robô siga linhas por de baixo de si e tem também um conjunto de quatro *true color RGB LEDs WS2812B*.

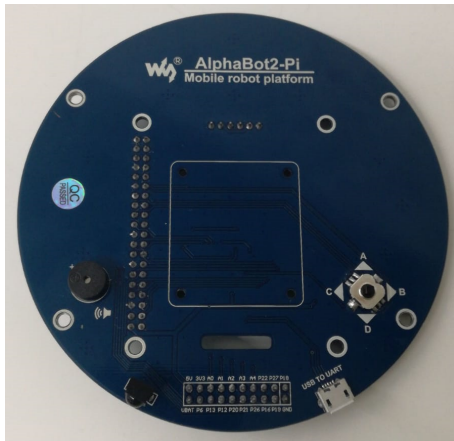


Figura 3.5: *AlphaBot2-Pi*: Vista da parte superior

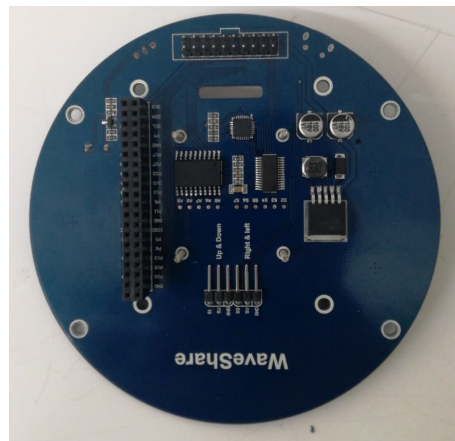


Figura 3.6: *AlphaBot2-Pi*: Vista da parte inferior

O *AlphaBot2-Pi*, tem na sua parte inferior 3.6 a interface que permite ligar ao *AlphaBot2-Base* e conectores GPIO que permitem ligar ao *Raspberry Pi*. Tem também pinos que permitem a ligação a dois motores servos que permitem rodar a câmara no eixo horizontal e vertical.

Na sua parte superior 3.5, este contém um *joystick*, um recetor de infravermelhos que permite o controlo remoto, um *buzzer* e orifícios destinados à instalação de uma câmara com o devido suporte para a mesma e dois servos, como pode ser observado na figura 3.7.

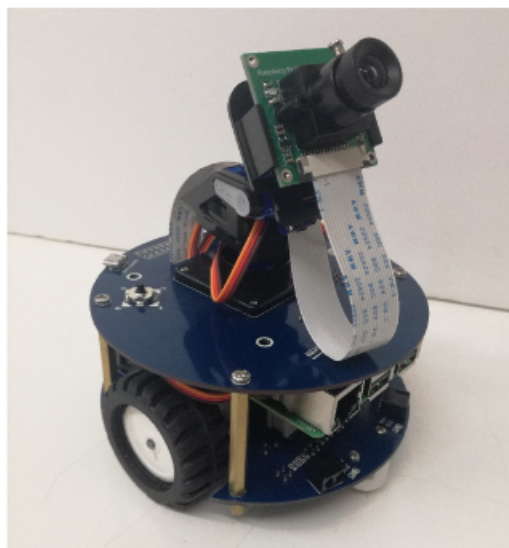


Figura 3.7: *AlphaBot2*

3.2.2.2 Percepção do espaço circundante

Os dois sensores infravermelhos que permitem a deteção de obstáculos e objetos na proximidade frontal do robô em duas direções, revelam ser bastante limitados. O facto de apenas ser possível saber que um objeto se encontra mais próximo do robô relativamente a uma distância calibrada previamente, torna o uso destes sensores uma opção inviável para os objetivos propostos. Aliado a esta limitação, o facto de apenas possuir dois sensores infravermelhos colocados na frente do robô, limita ainda mais o uso dos mesmos, pois não se possui informação sobre a retaguarda do robô. Acrescentar sensores, quer na frente ou na retaguarda, de modo a detetar obstáculos à frente, a trás, esquerda e direita é impraticável, visto que os pinos GPIO do *Raspberry Pi* já se encontram utilizados na totalidade.

Em alternativa ao uso dos sensores infravermelhos, pode-se utilizar a câmara para obter a percepção do espaço circundante recorrendo a métodos de análise e processamento de imagem para detetar obstáculos e outras características relevantes do espaço. A câmara na condição original do *AlphaBot2* é, no entanto, pouco útil, no sentido que apenas fornece informação sobre o campo de visão na frente do robô.

A figura 3.8, podemos observar um conjunto de circunferências concêntricas que estão distanciadas por cinco centímetros entre si, desde o centro até à circunferência de maior raio de cinquenta centímetros de raio, que foram usadas como referência para analisar as imagens obtidas em cada uma das opções estudadas à frente.



Figura 3.8: Circunferências usadas para referência

Câmara original

Na figura 3.9, podemos ver a imagem obtida com a câmara original do *Alphabot2*, que foi colocada na posição que permitisse ver o mais perto possível do robô, sendo que esta consegue apenas ver para a sua frente um raio num intervalo de aproximadamente quinze a vinte e cinco

centímetros e com um campo de visão horizontal muito reduzido que também não permite uma visão ampla do espaço circundante. O uso desta câmara está, portanto, invalidado pois não possui características que permitam a obtenção de uma imagem possível de ser analisada para efeitos de percepção do espaço circundante.

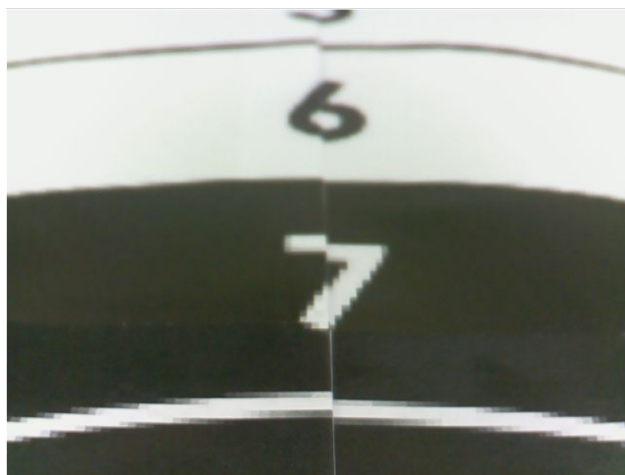


Figura 3.9: Vista da câmara original

A câmara original do *AlphaBot2*, a *RPi Camera (B)*, tem um campo de visão quer vertical (82.82°) quer horizontal (107°) reduzido quando comparado com a *RPi Camera (B) v2*, que possui uma maior abertura vertical (97.6°) e horizontal (124.4°). A segunda versão da câmara, também tem vantagem em relação ao tamanho da primeira, pois é achatada ao invés da primeira que tem uma lente mais volumosa.

Câmara com espelho convexo

Apenas utilizando uma câmara é impossível a percepção de todo o espaço circundante. Com o auxílio de um espelho, uma câmara pode captar uma imagem que contenha informação sobre um maior campo de visão, em particular, usando um espelho convexo, com o centro da câmara orientado ao centro do espelho.

Com a câmara numa posição horizontal centrada no meio do robô e com um espelho convexo por cima, podemos ter um campo de visão de 360° e assim ter uma perspetiva de todo o espaço circundante.

Experimentalmente, conclui-se que para o espelho disponível, a melhor altura para ser colocado, relativamente à parte superior do robô onde está colocada a câmara, seria de aproximadamente treze centímetros.

Recorrendo ao *software Autodesk Fusion 360*, foi criado um modelo CAD de um suporte composto por quatro peças individuais que servirão de suporte para o espelho à altura devida [3.10](#), fazendo uso dos orifícios e dos parafusos que seguram os *standoffs* do robô. Foi também criado

um suporte para a câmara [3.11](#), que será colocado no local destinado ao suporte da câmara original e que a mantenha na horizontal no centro do robô.

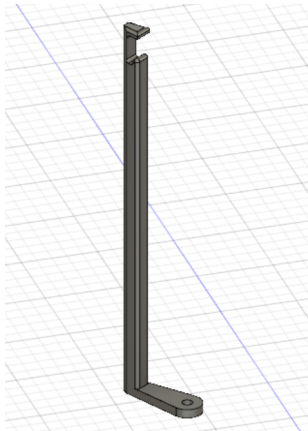


Figura 3.10: Suporte do espelho convexo

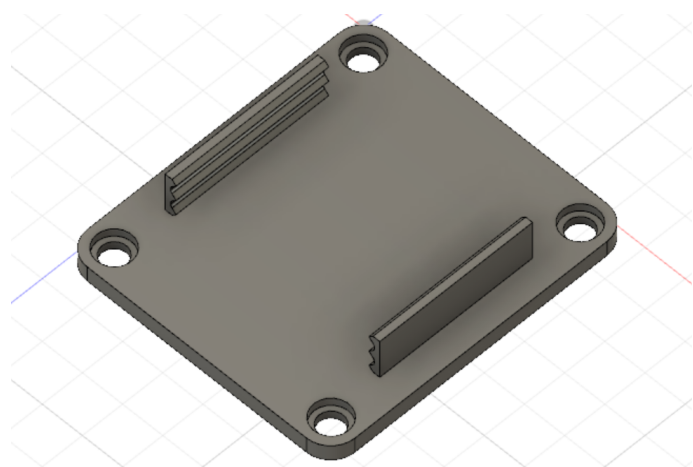


Figura 3.11: Suporte da câmara horizontal

O resultado da impressão 3D destes modelos CAD, pode ser visto nas seguintes figuras [3.12](#) e [3.13](#).



Figura 3.12: Suporte do espelho convexo

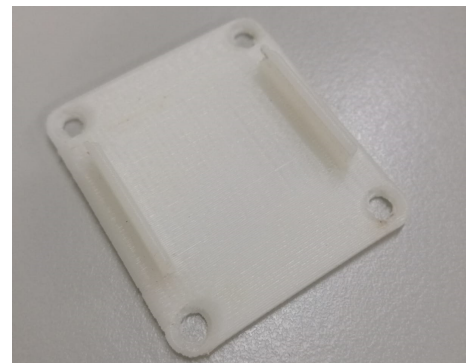


Figura 3.13: Suporte da câmara horizontal

O resultado da montagem das peças impressas no *AlphaBot2* e da consequente imagem captada pela câmara, podem ser observados nas figuras [3.14](#) e [3.15](#).

Pode-se observar na figura [3.15](#), que nas zonas próximas do robô, até cerca de vinte centímetros, há uma grande perceção do espaço sendo que esta está representada devido ao espelho convexo em maior destaque. Resultante disto, nas zonas mais afastadas do robô, a partir dos vinte centímetros, os detalhes da imagem foram bastante reduzidos, tornando-se mais dificultado o trabalho de analisar a imagem para a deteção de obstáculos e a que distância estes se encontram. Aliado a esta desvantagem, está também o facto da imagem ser distorcida não linearmente, o que dificulta ainda mais o processo de perceção do espaço circundante. A imagem obtida tem também

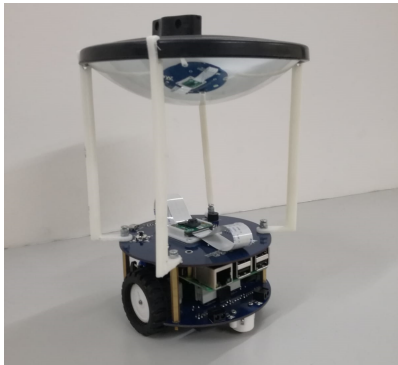


Figura 3.14: AlphaBot2 com o espelho convexo

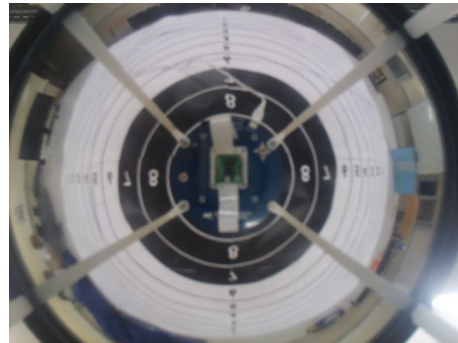


Figura 3.15: Imagem captada utilizando o espelho convexo

muitas áreas que podem ser desprezadas, sendo que os lados diretamente à esquerda e direita do robô não são fulcrais para o controlo do robô.

Este método revela, então, ser pouco viável, sendo necessária a utilização doutro tipo de espelho.

Câmara com espelhos em V

Em alternativa, com a câmara na mesma posição horizontal, mas ao invés da utilização de um espelho convexo que permita ver 360° à volta do robô, o uso de dois espelhos colocados em 'V' e orientados para a frente e para trás, torna-se uma opção plausível para obter uma imagem que seja possível de ser analisada para a deteção de obstáculos sendo capaz de:

- Ver para a frente e para trás;
- Ver uma distância até cinquenta centímetros;
- Ver o mais próximo possível do robô.

Para obter uma solução, é necessário saber a que altura deve ser colocado o espelho e qual o ângulo que este deve ser colocado. Para este efeito, é necessário usar um modelo geométrico do robô para calcular estes valores.

A figura 3.17, é um modelo geométrico da solução proposta, sendo que está representado um corte pelo plano XY perpendicular e simétrico ao eixo das rodas do robô representado na figura 3.16.

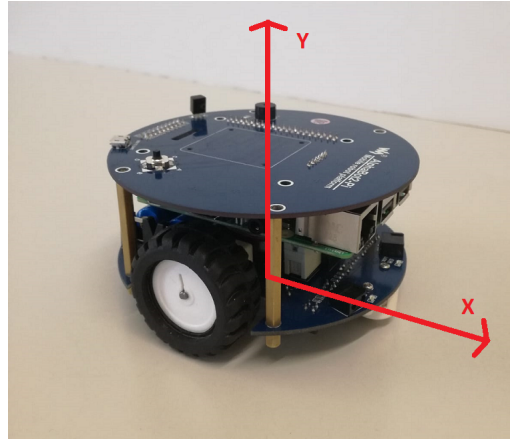


Figura 3.16: Plano XY sobre o qual foi feito o corte

Na figura 3.17 apenas são representadas as coordenadas positivas relativamente ao plano XY, sendo que o *AlphaBot2* é simétrico em relação ao eixo Y.



Figura 3.17: Modelo geométrico da solução com espelhos em 'V', no plano XY

Na figura 3.17:

- O espelho é representado pelo segmento de reta 'e' que une os pontos B e T;
- O *AlphaBot2* é representado pelo retângulo a azul que contem a Origem e o ponto A;
- Min e Max são as distâncias mínimas e máximas respetivamente que a imagem do robô consegue captar;
- i1 e i2 são os raios de incidência nas extremidades do espelho;

- r_1 e r_2 são os raios de refração nas extremidades do espelho;
- As retas p e q , são perpendiculares ao segmento de reta representado por BT .

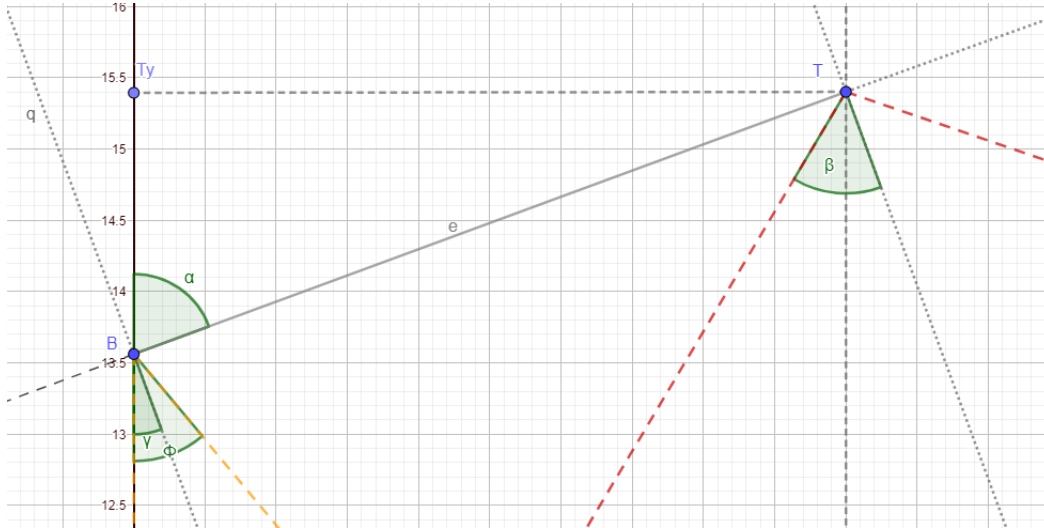


Figura 3.18: Representação geométrica 1

Na figura 3.18:

- O ângulo γ e β representam os ângulos de incidência de i_1 e i_2 respetivamente;
- O ângulo α é o ângulo que faz o espelho com o eixo Y que se pretende determinar;
- B e T são os pontos que representam a base e o topo do espelho respetivamente;
- O ângulo ϕ é formado pelas retas i_1 e r_1 no ponto B.

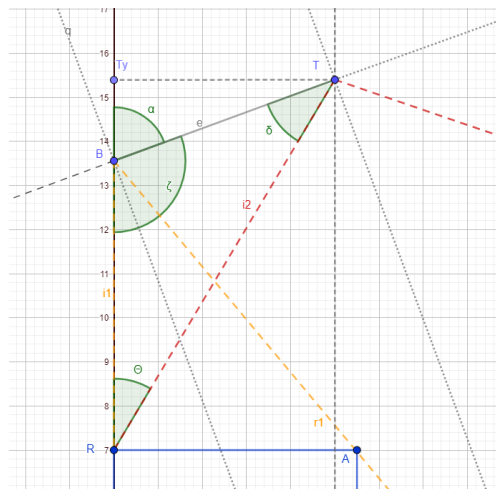


Figura 3.19: Representação geométrica 2

Na figura 3.19:

- O ângulo θ representa o ângulo de abertura da lente;
- R é o ponto que representa o centro da câmara;
- A distância entre os pontos R e B é a altura h variável que se pretende determinar.

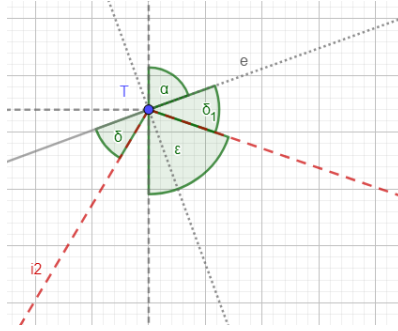


Figura 3.20: Representação geométrica 3

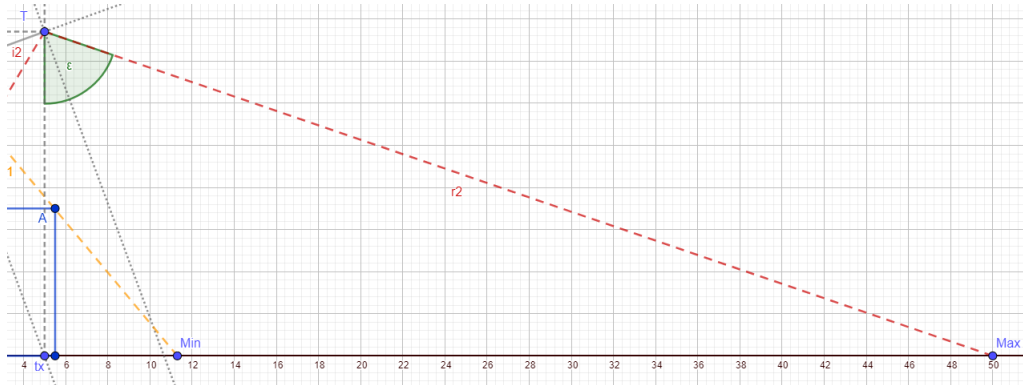


Figura 3.21: Representação geométrica 4

Um dos requisitos é que o robô seja capaz de detetar um obstáculo a cinquenta centímetros de distância, portanto Max terá esse valor. O ângulo θ é a abertura da lente, que para a câmara *RPi Camera (B) v2* assume um valor de 31.1° . É preciso, portanto, criar um sistema de quatro equações para determinar o valor das quatro variáveis α , e , h e Min , que minimize o valor de Min .

Segundo a figura 3.18, o valor de Min é mínimo quando as equações (3.5) e (3.6) se verificam:

$$\tan(\phi) = \frac{Min}{7+h} \quad (3.1)$$

$$\tan(\phi) = \frac{5.5}{h} \quad (3.2)$$

Também segundo a figura 3.18:

$$\gamma = 90^\circ - \alpha \quad (3.3)$$

$$\phi = 2 \cdot \gamma \quad (3.4)$$

Logo,

$$\Rightarrow \tan(2 \cdot (90^\circ - \alpha)) = \frac{Min}{7+h} \quad (3.5)$$

$$\Rightarrow \tan(2 \cdot (90^\circ - \alpha)) = \frac{5.5}{h} \quad (3.6)$$

Segundo a figura 3.21:

$$\tan(\varepsilon) = \frac{Max - e \cdot \sin(\alpha)}{h + 7 + e \cdot \cos(\alpha)} \quad (3.7)$$

Segundo a figura 3.19:

$$\zeta = 180^\circ - \alpha \quad (3.8)$$

$$\delta = 180^\circ - \theta - \zeta \quad (3.9)$$

Segundo a figura 3.20:

$$\delta = \delta_1 \quad (3.10)$$

$$\varepsilon = 180^\circ - \alpha - \delta_1 \quad (3.11)$$

Logo,

$$\Rightarrow \varepsilon = 211.1^\circ - 2 \cdot \alpha \quad (3.12)$$

Substituindo na equação 3.13:

$$\tan(211.1^\circ - 2 \cdot \alpha) = \frac{50 - e \cdot \sin(\alpha)}{h + 7 + e \cdot \cos(\alpha)} \quad (3.13)$$

Segundo a figura 3.19:

$$\tan(\theta) = \frac{e \cdot \sin(\alpha)}{h + e \cdot \cos(\alpha)} \quad (3.14)$$

Com as equações (3.5), (3.6), (3.13) e (3.14), podemos criar um sistema de quatro equações e quatro incógnitas:

$$\begin{cases} \tan(2 \cdot (90^\circ - \alpha)) = \frac{Min}{7+h} \\ \tan(2 \cdot (90^\circ - \alpha)) = \frac{5.5}{h} \\ \tan(211.1^\circ - 2 \cdot \alpha) = \frac{50 - e \cdot \sin(\alpha)}{h + 7 + e \cdot \cos(\alpha)} \\ \tan(31.1) = \frac{e \cdot \sin(\alpha)}{h + e \cdot \cos(\alpha)} \end{cases}$$

O resultado deste sistema possui várias soluções, no entanto no contexto em que está inserida, a solução a utilizar é:

$$\begin{cases} \alpha = 70^\circ \\ h = 65.6cm \\ e = 54.4cm \\ Min = 11.4cm \end{cases}$$

Recorrendo ao *software Autodesk Fusion 360*, foi criado um modelo CAD de um suporte composto por duas peças, com as devidas características determinadas através do sistema de equações anterior. Foi também criado um novo suporte para a câmara que permita estar na posição horizontal, no centro do robô, de modo a que o campo de visão horizontal da mesma, que é superior ao vertical, esteja orientado para ver a frente e a retaguarda do robô.

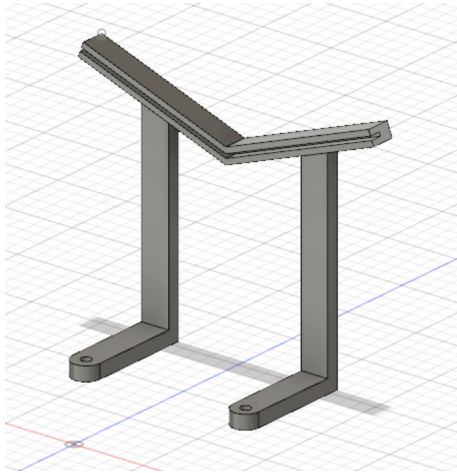


Figura 3.22: Suporte do espelho em 'V'

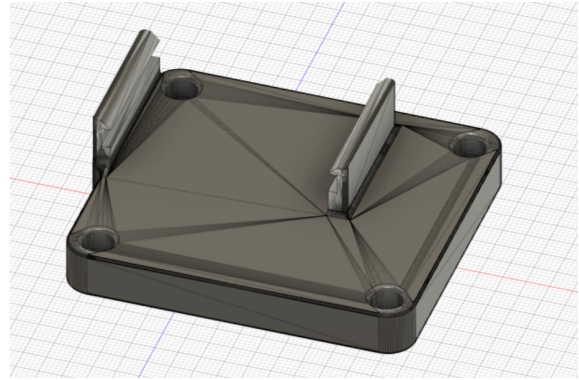


Figura 3.23: Suporte da câmara horizontal para o espelho em 'V'

O resultado das impressões 3D dos modelos CAD pode ser visto nas seguintes figuras 3.26 e 3.27.

O resultado da montagem das peças impressas no *AlphaBot2* e da imagem captada pela câmara, podem ser observados nas figuras 3.26 e 3.27.

Analisando a figura 3.27, podemos observar que a distância mínima captada pela câmara é aproximadamente o valor calculado anteriormente de 11.4 cm e que consegue captar um obstáculo a cerca de 45 cm de distância quer à frente, quer na retaguarda.

Na figura 3.28 podemos ver delimitado a vermelho, a área visível pelo *REDi3*, estando este representado a azul com o suporte dos espelhos a laranja. Podemos observar que, tanto à frente,



Figura 3.24: Suporte do espelho em 'V'

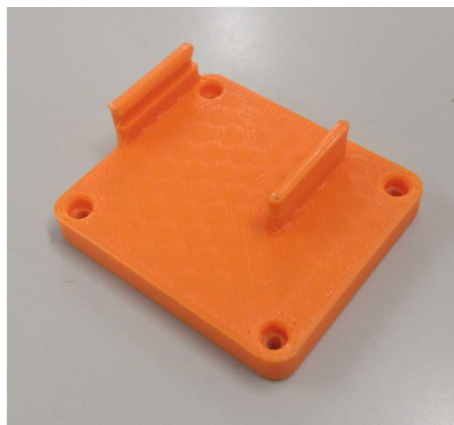


Figura 3.25: Suporte da câmera horizontal para o espelho em 'V'

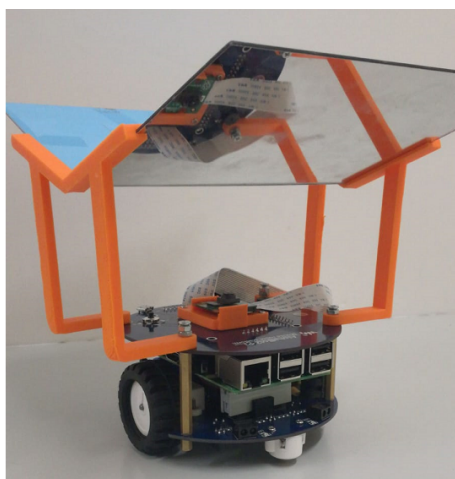


Figura 3.26: *AlphaBot2* com os espelhos em 'V'



Figura 3.27: Imagem captada usando os espelhos em 'V'

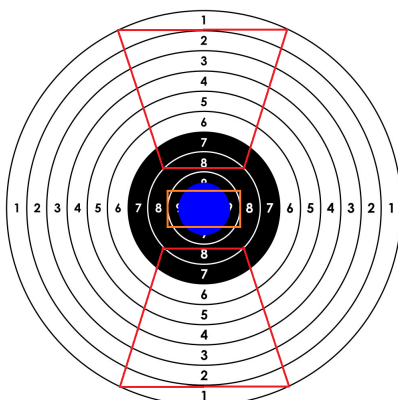


Figura 3.28: Representação do Espaço Visível

como à retaguarda do robô, o espaço detetado pela câmara na zona mais próxima do *REDi3*, apesar de ser reduzido, com uma abertura de cerca de 35 graus, é suficientemente largo para que este não falhe a deteção de objetos nestas direções. O espaço entre a área a vermelho e o robô na parte frontal que não é captado pela câmara, pode ser detetado pelos dois sensores infravermelhos originais do *Alphabot2* calibrados para aquela distância, que garantem que exista deteção de objetos naquela zona.

3.2.2.3 Controlo do *REDi3*

Na figura 3.29, pode-se observar um diagrama no qual estão representados os vários módulos presentes nesta solução do problema.

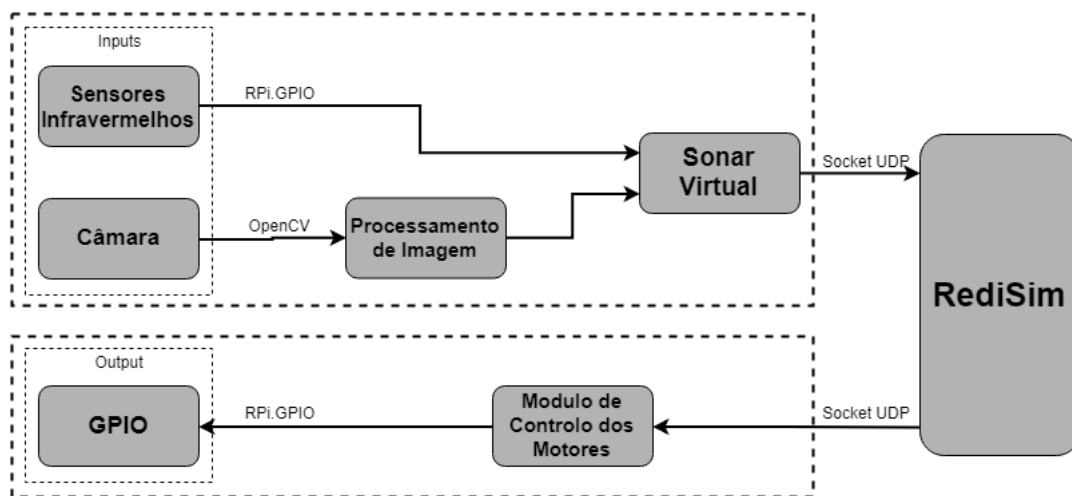


Figura 3.29: Diagrama de controlo

As entradas do sistema são os sensores infravermelhos e a imagem fornecida pela câmara. O valor dos sensores infravermelhos obtidos usando a biblioteca *RPi.GPIO* e o resultado do processamento da imagem em *OpenCV*, são utilizados para a criação de sonares virtuais.

O *RediSim*, recebe via *Socket UDP* o valor dos sonares e efetua a simulação do comportamento do robô para estes valores retornando, também através de *Socket UDP*, valores que correspondem à velocidade que os motores devem atuar.

Esses valores são convertidos, através de um modulo de controlo dos motores e transformados em saídas do GPIO correspondentes aos mesmos.

3.2.2.4 Processamento da Imagem

Na linguagem de programação *Python*, é possível fazer uso das bibliotecas de *OpenCV 2.5.3* para ter acesso às imagens capturadas pela câmara e proceder ao processamento e análise das mesmas.

Na figura 3.30, pode-se ver uma imagem capturada pela câmara no estado original e na figura 3.31, estão representadas as áreas sob as quais a imagem original será dividida em quatro imagens

diferentes. A área 1, representa a secção da imagem correspondente à parte de trás do *REDi3* e nas áreas 2, 3 e 4, são as secções correspondentes à parte da frente, sendo respetivamente a esquerda, o meio e a direita.

Podemos ver no centro da figura 3.30, uma linha na vertical, resultante da junção dos dois espelhos em 'V', que na figura 3.31 está representada a vermelho. Esta secção da imagem é removida e não faz parte de nenhuma área de interesse da imagem, visto não fornecer nenhuma informação sobre o espaço circundante.



Figura 3.30: Exemplo de imagem capturada

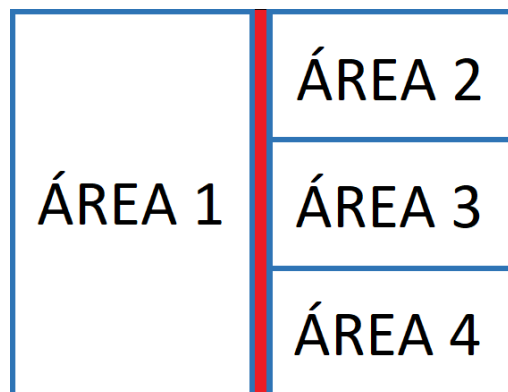


Figura 3.31: Divisão da imagem por zonas

Após a divisão da imagem da figura 3.30, podemos ver na figura 3.32, a imagem resultante para a área 1. Para cada uma das áreas, será aplicado o mesmo processo descrito abaixo.



Figura 3.32: Área 1 da
imagem original



Figura 3.33: Tons de
cinzento



Figura 3.34: Imagem
binária



Figura 3.35: Contornos
da Área 1

Começa-se por aplicar uma mudança do *color-space* da imagem, sendo que é alterado para uma imagem em tons de cinzento em que a cada pixel da imagem corresponde um valor entre 0 e 255, sendo que os valores mais próximos de 0 correspondem a tons mais escuros e os próximos de 255 correspondem a tons mais claros. O resultado desta alteração pode ser observado na figura 3.33. É de notar que as partes da imagem que correspondem ao *background* assumem tons mais claros, e a mão representada na imagem assume tonalidades mais escuras.

À imagem em tons de cinzento da figura 3.33, aplica-se uma binarização utilizando um valor de *threshold*, ao qual seja possível separar o fundo claro da imagem e os obstáculos mais escuros e obter uma imagem binária. O resultado pode ser observado na figura 3.34, na qual o fundo se encontra a preto e os obstáculos a branco e se consegue notar com clareza os pontos da imagem que separam o *background* e o obstáculo.

A esta imagem binária, é aplicada uma função do *OpenCV* que encontra os pontos que representam a fronteira entre o preto e o branco. Esta função retorna um vetor que contém todos os contornos identificados na imagem e a cada um contém outro vetor com o conjunto de pontos que são contornos do obstáculo. A representação dos contornos pode ser observada na figura 3.35.

Aplicando o método descrito em cima para todas as imagens resultantes da separação da imagem original, obtemos o resultado que pode ser observado nas figuras 3.36, 3.37, 3.38 e 3.39.



Figura 3.36: Contornos da retaguarda

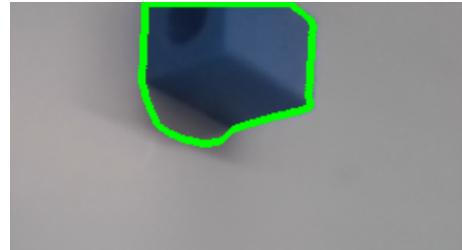


Figura 3.37: Contornos da direita

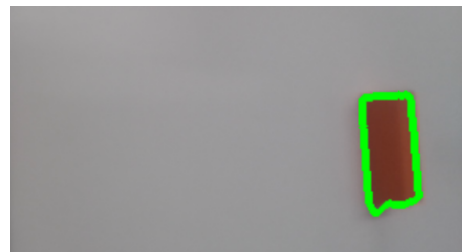


Figura 3.38: Contornos do meio



Figura 3.39: Contornos da esquerda

Para cada uma das imagens, existe o vetor dos contornos associados a esta. Fazendo uma busca pelo vetor de modo a determinar qual o ponto dos contornos que está mais próximo do centro da imagem, podemos descobrir qual a distância em pixels a que obstáculo se encontra.

3.2.3 Sonares Virtuais

Utilizando como referencia a imagem da figura 3.27, e sabendo que as circunferências são equidistantes umas das outras com um valor de 5 cm, reúne-se um conjunto de pontos para cada uma das áreas descritas na figura 3.31.

Para cada área tomou-se como referencia uma linha horizontal no centro da imagem e retiraram-se os dados da distância e do pixel correspondente à interseção desta linha com a linha da circunferência.

Com estes dados é possível criar uma equação aproximada para cada uma das áreas, de modo a determinar para cada pixel, a distância real em centímetros a que aquele ponto se encontra. Na figura 3.40, podemos ver os pontos recolhidos da figura 3.27 para cada área e a respetiva linha de tendência.

Utilizando para cada área a respetiva linha de tendência, é possível determinar a distância aproximada da realidade que um ponto se encontra do REDi3.

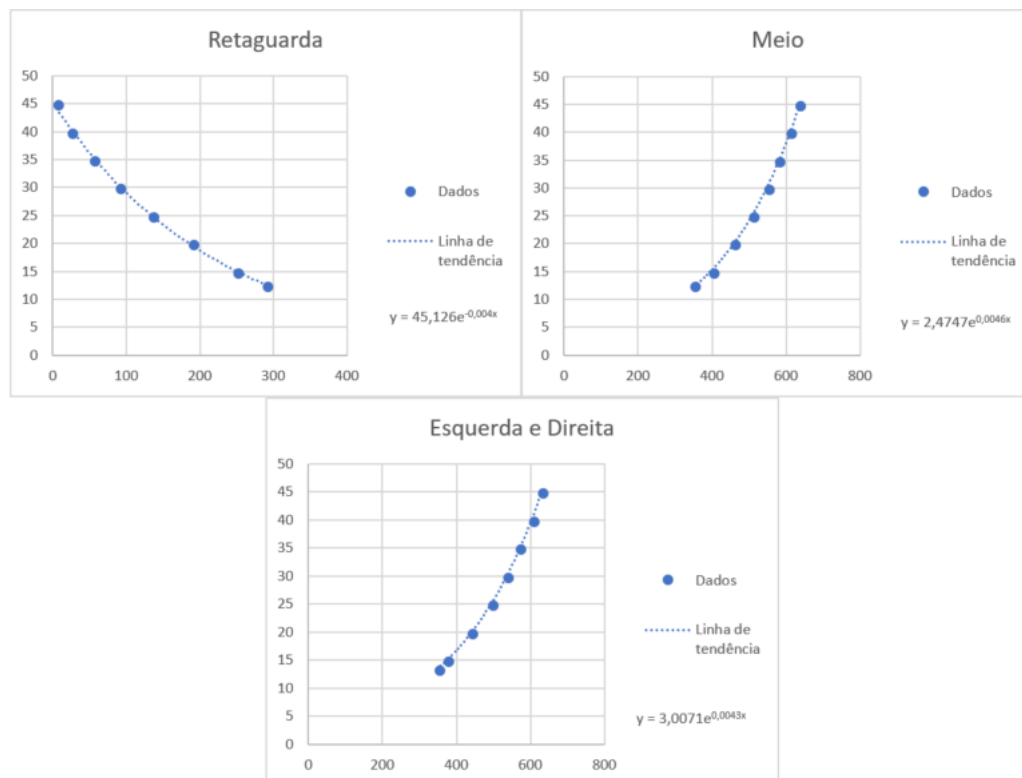


Figura 3.40: Linhas de tendência

Com os dados obtidos em cima, apenas é possível determinar distâncias para objetos presentes entre os limites determinados anteriormente (de 11.4 cm a 45 cm).

Calibrando os sensores infravermelhos originais do *Alphabot2* para detetar a presença de um objeto até ao limite mais próximo do REDi3 na imagem obtida, podemos obter a informação se existe algum objeto naquela zona que a câmara não seja capaz de detetar.

Utilizando os dados obtidos em cima, é efetua-se uma escala percentual da distancia a que um objeto se encontra, para que esta informação possa ser utilizada como um sonar virtual. O resultado desta operação pode ser observado na figura 3.41.

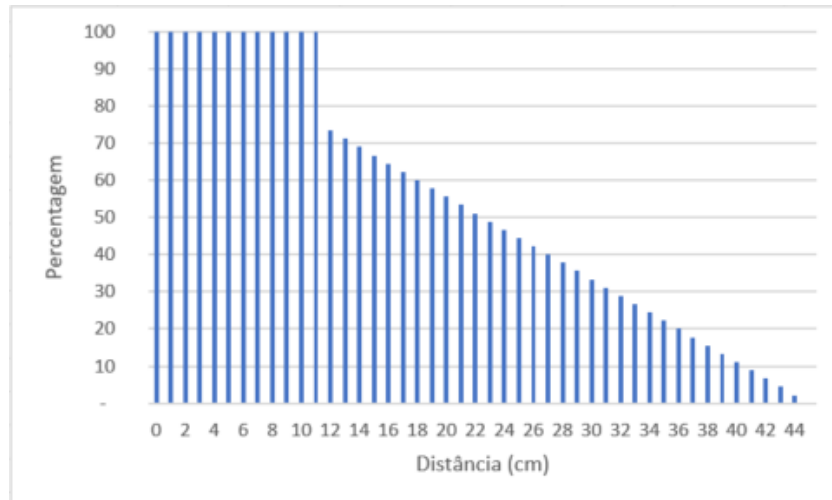


Figura 3.41: Gráfico percentual da distância

Com este método, o sonar virtual envia valores periodicamente para o RediSim com uma periodicidade de cerca de 50 milissegundos.

3.2.4 Utilização do Simulador *RediSim*

Através de uma ligação efetuada pelo protocolo de comunicação *VNC* feita a partir de um *tablet* ou outro dispositivo *handheld* ao *Raspberry Pi*, o utilizador insere na interface gráfica do *RediSim* 2.5.1.1 a ligação das cavilhas virtuais desejadas para o problema em causa.

O utilizador pode também escolher utilizar o *RediSim* como simplesmente um simulador, ou pode escolher usa-lo para o efeito de ser utilizado com uma ferramenta do *REDi3*. Para tal, foram feitas alterações de modo a que o simulador receba os valores dos sonares virtuais e retorne os valores correspondentes à velocidade dos motores.

O simulador está repetidamente a receber mensagens da *Socket UDP* que contém os valores dos sonares virtuais, ao invés de utilizar os dados dos sonares da simulação num mapa virtual 2.9 criado pelo utilizador.

O *RediSim*, conforme a configuração da interface das cavilhas 2.8 inserida pelo utilizador e os valores dos sonares virtuais, simula o valor da velocidade dos motores para o estado atual. Estes são enviados via *Socket UDP* para o Modulo de Controlo dos Motores.

3.2.5 Modulo de Controlo dos Motores

Na linguagem de programação *Python*, este modulo está constantemente a receber mensagens via *Socket UDP* do valor da velocidade dos motores calculado e enviado pelo *RediSim*.

Utiliza-se as bibliotecas *RPi.GPIO* e *Alphabot2*, que permitem a alteração das saídas do *Raspberry Pi* para os valores de PWM pretendidos.

O valor de PWM de cada motor e o seu sentido, é calculado usando o valor recebido via *Socket UDP* do valor da velocidade obtido no *RediSim*.

3.2.6 Interface, Registo e Consulta de Robôs

O REDi3 necessita da interação do utilizador por via de uma interface gráfica de modo a inserir no RediSim a configuração desejada das cavilhas das ligações virtuais.

De modo a que seja possível a visualização da interface do *RediSim*, sem necessidade de recurso à utilização de um monitor local para a representação do ambiente gráfico do *Raspberry Pi*, destacam-se duas abordagens possíveis: utilização de um ambiente gráfico remoto com recurso a VNC que permite a utilização num dispositivo *handheld*, figura 3.42; ou a utilização do protocolo de rede SSH, figura 3.43, que permite a conexão de um cliente com um servidor, acesso a uma execução remota de comandos e por sua vez permite a visualização da interface gráfica do *RediSim*.

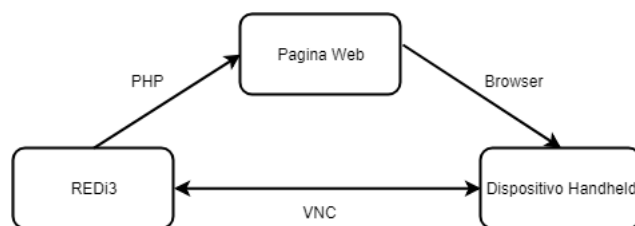


Figura 3.42: Interface com recurso a VNC

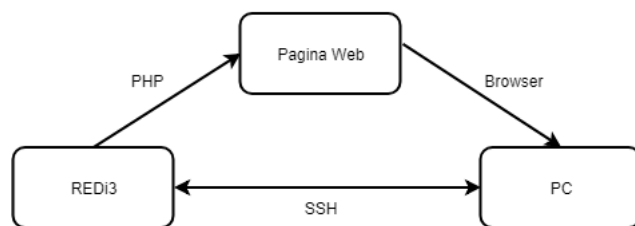


Figura 3.43: Interface com recurso a SSH

Em ambas opções, é necessário o cliente saber o endereço IP do servidor, que no caso da rede *Eduroam*, se altera a cada conexão e não é possível a fixação do mesmo num endereço conhecido, sem o recurso a entidades capazes de o atribuírem.

De modo a que o cliente seja capaz de se conectar utilizando qualquer um dos protocolos, é utilizado um ficheiro em PHP que é executado automaticamente quando o REDi3 é ligado. Este ficheiro escreve numa pagina web o endereço IP que lhe está atribuído, de modo a que o cliente possa efetuar a ligação.

Do lado do cliente Redi3, o seguinte código bash shell script é executado:


```
#!/bin/bash
OUT=`ifconfig wlan0 | grep "inet addr" | cut -d ':' -f 2 | cut -d ' ' -f 1`
echo `curl -X POST https://paginas.fe.up.pt/~asousa/redi3/register.php?log=$OUT`
```

Do lado do PHP, para o registo do robô:

```
if (isset($_GET['log'])) {
    $payload = $_GET['log']; //recebe parâmetro do URL (via get ?log=xxxx)
} else {
    echo "Nothing to ?Log= <br/>";
    exit(0);
}

$datetime = date("Y/m/d , , H:i:s , , "); //função PHP que retorna a data atual
$date_fname = date("Y_m_d");
$line = $datetime.$_SERVER['REMOTE_ADDR']." , , $payload \r\n";

$name="redi3".$date_fname.".txt";
if (!$file = fopen($name, 'a+')) {
    echo "Erro a abrir o ficheiro $name <br/>";
    exit;
}

while(flock($file, LOCK_EX) == FALSE && (contador < 10) ) { // tenta fazer lock 10 vezes
    $contador++;
    usleep (50000); // 50 ms
    if ($contador==9)
        break;
}

if (($contador < 10) && flock($file, LOCK_EX) == TRUE) {
    if (!fwrite($file, $line)){
        echo "Erro na escrita do ficheiro \r\n";
    }
} else {
    flock($file, LOCK_UN); // libera o lock
    echo "Não consegui fazer lock <br/>";
}
fclose($file);
```

Para ler o registo dos robôs que se registaram no dia de hoje, passado como parâmetro GET do PHP:

```
$ficheiro_registro = $_GET['log'];
//Variável $file armazena a conexão com o arquivo e o tipo de ação (read).
if (!$file = fopen($ficheiro_registro, 'r')) {
    echo "Erro a abrir o ficheiro para leitura <br/>";
    exit;
}

while (!feof ($file)) {
    $valor = fgets($file, 4096);

    echo $valor."<br>";
}
fclose($file);
```

3.2.7 Utilização da Câmara Original do AlphaBot2

No ponto 3.2.2.2 Perceção do espaço circundante, foi estudado o uso da câmara original do AlphaBot2 e conclui-se que para o desenvolvimento deste projeto, esta deveria ser substituída pela versão seguinte da mesma, devido as suas limitações de captura de imagem.

Após o processo de desenvolvimento do projeto utilizando a *RPi Camera (B) v2*, repetiu-se o processo do cálculo das características dos suportes dos espelhos e do espelho em si, nomeadamente o comprimento dos espelhos, o ângulo entre eles e a altura a que devem ser colocados.

O resultado destas alterações pode ser visto na seguinte na figura 3.44.

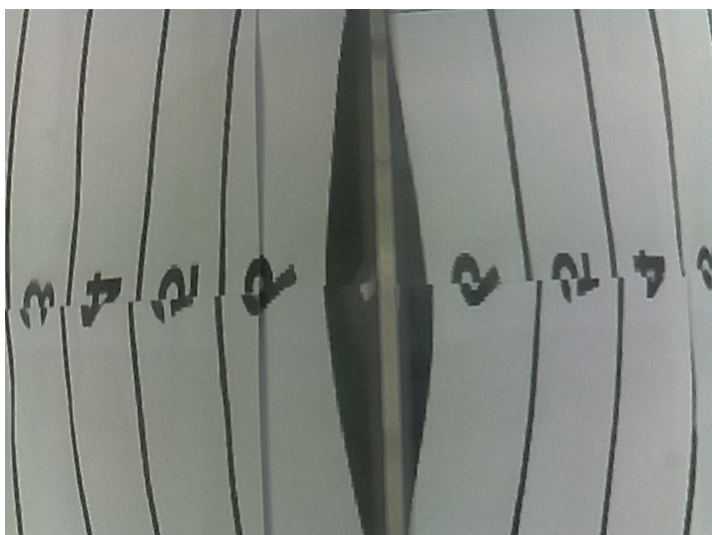


Figura 3.44: Imagem captada com a configuração para a câmara original

Podemos ver na figura, que o resultado obtido no que diz respeito às distâncias a que a a câmara consegue captar a imagem que o alcance máximo foi reduzido para cerca de quarenta centímetros em cada direção e o alcance mínimo é de cerca de dezassete centímetros.

A utilização desta câmara apresenta algumas limitações na percepção do espaço circundante, pois possui uma área mais limitada sobre a qual é capaz de captar imagem. A alteração das distâncias que o REDi3 consegue ver utilizando esta câmara, torna também necessário a alteração do ponto 3.2.3 Sonares Virtuais para que o restante do sistema possa funcionar em condições semelhantes à utilização da *RPi Camera (B) v2*.

3.2.8 Bateria

O AlphaBot2 possui um suporte para baterias 14500 de 3.7 V. A utilização deste tipo de baterias revelou não ser suficiente, visto possuir uma autonomia de cerca de 15 minutos de utilização. A substituição de baterias também se revelou não ser satisfatória, visto que para efetuar a troca por baterias carregadas, é necessário proceder a uma desmontagem de um numero de partes considerável do robô que torna inviável para a utilização em demonstrações.

Foi portanto desenhado um novo suporte de baterias capaz de receber baterias 18650, que permite ao REDi3 ter uma autonomia de cerca de uma hora. Este suporte foi também modelado de forma a que não esteja fixo numa posição do robô e que fosse possível efetuar a troca de baterias de um modo que não implique a desmontagem de nenhuma parte do robô e torne a substituição um processo mais simples e rápido.

O resultado da impressão 3D representado na figura 3.47 pode ser visto nas figuras 3.45 e 3.46, em que podemos ver a capacidade deste suporte de ser removido de modo a facilitar a substituição de baterias.

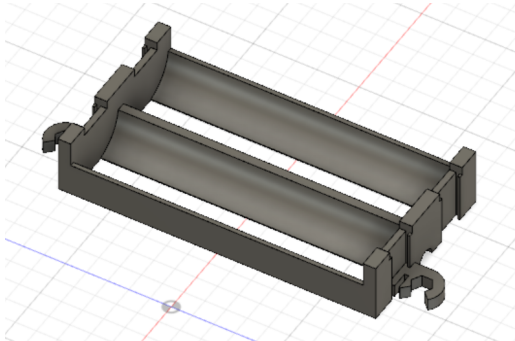


Figura 3.45: Suporte das baterias

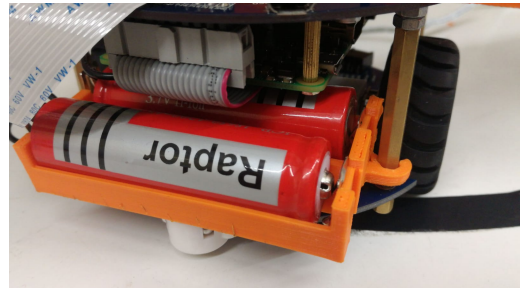


Figura 3.46: Suporte colocado

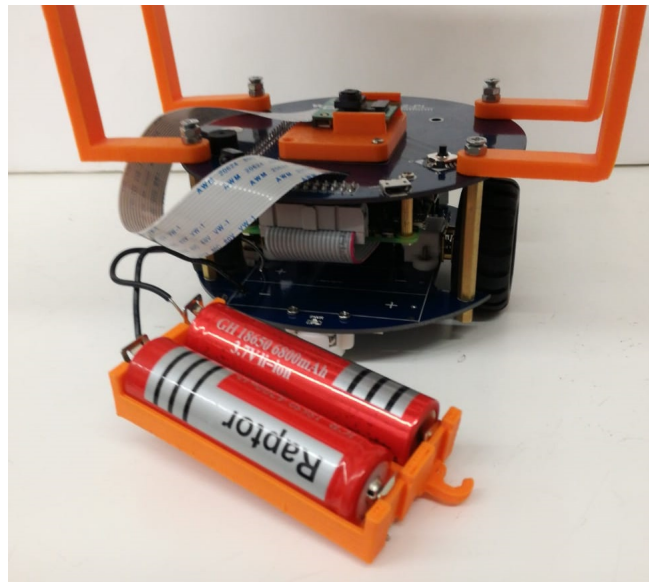


Figura 3.47: Suporte em pose de substituição

Capítulo 4

Conclusões

Esta dissertação procura construir um robô para educação e demonstrações que não deve necessitar de apoio de um PC para a sua programação. No seguimento de trabalhos anteriores "REDi", procura-se um robô que permita programação por fios virtuais e que não necessite obrigatoriamente de um PC externo para a sua programação. Pretende-se ainda um sistema com componentes genéricos e de custo baixo mas com a flexibilidade para desenvolvimentos futuros.

Nesta dissertação foi estudada uma proposta de solução chamada REDI2 que seria a construção de um robô com recursos disponíveis com facilidade mas esta solução apresenta preço na ordem dos 300 euros (a custo mínimo, à data da dissertação, 2018).

O *REDi3* é uma solução capaz de dar resposta aos desafios impostos por intermédio da adaptação e melhoramento do robô Waveshare AlphaBot 2 na versão para Rapsberry Pi.

O custo deste robô é interessante uma vez que o kit *AlphaBot2* custa cerca de 100 euros incluindo a câmara mas sem o RPi.

Os melhoramentos mecânicos recorrem a impressões 3D de custo muito reduzido (estima-se 1 euro de consumível plástico PLA) e espelho plástico acrílico, leve, seguro e fácil de trabalhar com o custo aproximado de de 5 euros.

O Redi3 recorre a processamento de imagem para obter um robô com sonares virtuais via processamento de imagem, supondo chão claro e objetos contrastantes.

Foram estudadas duas câmaras mas a câmara original incluída no AlphaBot2 é utilizável, apesar de ter uma menor perceção do espaço circundante.

Foram ainda produzidos suportes para baterias de iões lítio 18650 de capacidade muito superior às originalmente propostas.

O robô pode ser acedido via *tablet* em redes anteriormente configuradas pois auto regista-se numa página publicamente visível.

O software de visão desenvolvido liga à biblioteca do fornecedor utilizando a lógica de ligação pré existente no *software RediSim*.

O conjunto foi testado e apresenta-se funcional apesar do robô se mover lentamente devido ao processamento de imagem não ser rápido e limitações de comunicações.

Todos os componentes utilizados são abertos para facilitar desenvolvimentos futuros. A utilização do R-Pi é vantajosa pois certamente aparecerão outras variantes com desempenho superior e o hardware ser standard facilita desenvolvimentos futuros, incluindo novas formas de programação.

Com a realização deste trabalho de dissertação, foi feita uma contribuição para o Departamento de Engenharia Eletrotécnica e de Computadores, no sentido de ter sido criado um robô que pode ser replicado a custo relativamente baixo, para ser utilizado em demonstrações do departamento.

Anexo A

Redi2 - Custos dos Componentes

Tabela A.1: Preço dos componentes da Solução Proposta

Componente	Quantidade	Preço/Un. (EUR)	Local de compra
Modulo de câmara para Raspberry Pi V2 8MP 1080p	1	30.45	http://www.botnroll.com/pt/camaras/1815-modulo-de-camara-para-raspberry-pi-v2-8mp-1080p-.html
Seguidor de Linha QTR-8A	1	8.02	https://www.pololu.com/product/960
Raspberry Pi3	1	43.98	https://mauser.pt/catalog/product_info.php?products_id=74332&src=raspberry
Arduino Uno	1	14.90	http://www.botnroll.com/pt/arduino-controladores/2298-arduino-uno-compativel-uno-plus.html?search_query=arduino+uno&results=131
100:1 Micro Metal Gearmotor Mp 6V	2	15.95	https://www.pololu.com/product/2130
DRV8833 Dual Motor Driver Carrier	1	4.95	https://www.pololu.com/product/2130
SRF05 - Ultrasons	4	18.70	http://www.botnroll.com/pt/sonares/114-srf05.html?search_query=sonar&results=20
Módulo OLED 0.96	1	9.90	http://www.botnroll.com/pt/lcds-e-displays/2028-modulo-oled-096-b-.html
Rodas	2	3	http://www.botnroll.com/pt/rodas/2164-par-de-rodas-de-borracha-65mm.html?search_query=rodas&results=37
Standoffs	8	-	
Placa PVC 20x20x2mm	3	-	
Powerbank 2200mAh	3	24.60	https://www.ptrobotics.com/lipo/2396-2200mah-li-ion-rechargeable-power-bank.html?search_query=power+bAnk+&results=18
Total		295,70	

Anexo B

Redi3 - Custos dos Componentes

Componente	Quantidade	Preço/Un(Euros)	Local de Compra
R-Pi 3B	1	40	https://www.aquario.pt/pt/product/raspberry-pi-3-modelo-b-1gb
Kit AlphaBot2	1	100	https://www.amazon.es/Alphabot2-CQRobot-Raspberry-Functions-Monitoring/dp/B077J5GQD3/ref=sr_1_4?ie=UTF8&camp;qid=1517
Espelho	1	10	https://www.amazon.es/gp/product/B0074GOLH6/ref=ox_sc_act_title_2?smid=AZBN1KQ5H33U8&psc=1
Total		150	

Referências

- [1] First LEGO League. URL: <http://www.firstlegoleagueportugal.org/first-lego-league-portugal>.
- [2] RoboParty. URL: <http://www.roboparty.org/abstract.php>.
- [3] RoboParty Race of Champions. URL: <http://www.roboparty.org/race.php>.
- [4] RoboParty Obstáculos. URL: <http://www.roboparty.org/obst.php>.
- [5] RoboParty Dança. URL: <http://www.roboparty.org/dance.php>.
- [6] Festival Nacional de Robotica. URL: <https://robotica2018.festivalrobotica.pt/pt>.
- [7] RoboCup. URL: <http://www.robocup.org/>.
- [8] ICARSC. URL: <http://robotica2018.festivalrobotica.pt/pt/icarsc>.
- [9] Universidade Junior. URL: <https://universidadejunior.up.pt/>.
- [10] Stefano Carpin, Mike Lewis, Jijun Wang, Stephen Balakirsky, e Chris Scrapper. USAR-Sim: A robot simulator for research and education. Em *Proceedings - IEEE International Conference on Robotics and Automation*, 2007. doi:10.1109/ROBOT.2007.363180.
- [11] Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klapotcz, Stéphane Magnenat, Jean-Christophe Zufferey, Dario Floreano, e Alchero Martinoli. The e-puck, a Robot Designed for Education in Engineering.
- [12] Cam Jam. URL: http://camjam.me/?page_id=1035.
- [13] AlphaBot 2 User Manual 1 AlphaBot 2 User Manual. 2017.
- [14] Filipe Lopes, Hugo Costa, e Susana Neves. REDI Robot de Ensino Didático.
- [15] Bruno Pires. Plataforma Robótica Multi-Funcional.
- [16] Legomindstorm EV3. URL: <https://www.lego.com/en-us/mindstorms/products/mindstorms-ev3-31313>.
- [17] Makeblock mbot v1.1. URL: <http://store.makeblock.com/product/mbot-robot-kit>.